

TUGAS AKHIR - KS 141501

**PENERAPAN METODE *GROUP METHOD OF DATA HANDLING* (GMDH) *NETWORK* PADA PERAMALAN DATA INTERMITEN**

***APPLICATION OF A GROUP METHOD OF DATA HANDLING (GMDH) NETWORK METHOD FOR INTERMITTENT DATA***

ZUHRIYA FIRDA  
NRP. 5214100138

Dosen Pembimbing:  
Wiwik Anggraeni, S.Si., M.Kom.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





TUGAS AKHIR - KS 141501

**PENERAPAN METODE *GROUP METHOD OF DATA HANDLING (GMDH) NETWORK* PADA PERAMALAN DATA INTERMITEN**

ZUHRIYA FIRDA  
NRP. 5214100138

Dosen Pembimbing:  
Wiwik Anggraeni, S.Si., M.Kom.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





FINAL PROJECT - KS 141501

***APPLICATION OF A GROUP METHOD OF DATA  
HANDLING (GMDH) NETWORK METHOD FOR  
INTERMITTENT DATA***

ZUHRIYA FIRDA  
NRP. 5214100138

Supervisor:  
Wiwik Anggraeni, S.Si., M.Kom.

INFORMATION SYSTEMS DEPARTEMENT  
Information and Communication Tecnology Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



## LEMBAR PENGESAHAN

### **PENERAPAN METODE *GROUP METHOD OF DATA HANDLING* (GMDH) NETWORK PADA PERAMALAN DATA INTERMITEN**

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**Zuhriya Firda**  
**5214100138**

Surabaya, 16 Januari 2018

**Pih KEPALA**  
**DEPARTEMEN SISTEM INFORMASI**

**Edwin Riksakomara, S.Kom, M.T.**  
**NIP.196907252003121001**







## LEMBAR PERSETUJUAN

### **PENERAPAN METODE *GROUP METHOD OF DATA HANDLING* (GMDH) *NETWORK* PADA PERAMALAN DATA INTERMITEN**

#### **TUGAS AKHIR**

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**ZUHRIYA FIRDA**  
**NRP. 05211440000138**

Disetujui Tim Penguji : Tanggal Ujian : 12 Januari 2018  
Periode Wisuda: Maret 2018

**Wiwik Anggraeni, S.Si., M.Kom.**

  
(Pembimbing 1)

**Edwin Riksakomara, S.Kom., M.T.**

  
(Penguji 1)

**Faizal Mahananto, S.Kom., M.Eng., Ph.D.**

  
(Penguji 2)



# **PENERAPAN METODE *GROUP METHOD OF DATA HANDLING* (GMDH) NETWORK PADA PERAMALAN DATA INTERMITEN**

**Nama Mahasiswa** : Zuhriya Firda  
**NRP** : 5214100138  
**Jurusan** : Sistem Informasi FTIK-ITS  
**Dosen Pembimbing** : Wiwik Anggraeni S.Si, M.Kom

## **ABSTRAK**

*Dalam mengelola rantai pasok pada suatu organisasi atau perusahaan, permasalahan umum yang terjadi adalah ketidakpastian permintaan. Pola permintaan merupakan hal penting yang perlu diperhatikan untuk dapat melakukan perencanaan dan pengelolaan rantai pasok secara efektif. Untuk menangani permasalahan ketidakpastian permintaan tersebut adalah melakukan peramalan permintaan. Namun tidak semua metode peramalan dapat memberikan tingkat keakuratan yang lebih baik terutama untuk pola permintaan yang tidak teratur dan intermiten. Pola data intermiten adalah pola data permintaan yang pada beberapa periode tidak ada permintaan sehingga permintaan terjadi secara terputus-putus. Hal ini akan sulit diramalkan karena data yang dalam banyak periode mengalami permintaan nol tersebut tidak dapat diselesaikan dengan menggunakan metode deret waktu konvensional seperti moving average, exponential smoothing, regresi linier, dan lain sebagainya. Hasil yang diperoleh dengan menggunakan metode tersebut akan menghasilkan tingkat kesalahan yang tinggi.*

*Metode yang digunakan pada tugas akhir ini untuk menangani masalah tersebut adalah menggunakan jaringan GMDH (*Group Method of Data Handling*) dimana metode ini dinilai pada penelitian-penelitian sebelumnya dapat menyelesaikan permasalahan permintaan yang intermiten dan memberikan*

nilai akurasi yang lebih baik jika dibandingkan dengan metode lainnya. GMDH merupakan metode dengan stuktur jaringan syaraf tiruan yang menggunakan fungsi transfer polinomial dan dapat membangun model secara mandiri dengan melakukan estimasi berdasarkan Root Mean Square Error (RMSE) dan melakukan penambahan lapisan pada hidden layer hingga didapatkan hasil keluaran yang terbaik.

Hasil analisa menunjukkan bahwa model jaringan GMDH dengan jumlah neuron input sebanyak 6, jumlah maksimum neuron 15, jumlah hidden layer sebanyak 20, dan nilai selection pressure 0,2 adalah model stuktur jaringan yang terbaik untuk meramalkan data intermiten. Nilai RMSE yang diperoleh yaitu senilai 14861 selama proses pelatihan dan 21721 untuk data pengujian dimana angka tersebut lebih kecil jika dibandingkan model lainnya. Selain itu, analisa MDA menunjukkan bahwa peramalan yang dihasilkan dari model tersebut dapat mengikuti pola data aktual dengan baik. Hal ini dibuktikan dengan nilai MDA 0,7983 untuk pelatihan dan 0,7222 untuk pengujian.

**Kata kunci:** Peramalan Data Intermiten, Permintaan Nol, Group Method of Data Handling (GMDH)

# **APPLICATION OF A GROUP METHOD OF DATA HANDLING (GMDH) NETWORK METHOD FOR INTERMITTENT DATA**

**Name** : Zuhriya Firda  
**NRP** : 5214100138  
**Departement** : Sistem Informasi FTIK-ITS  
**Supervisor** : Wiwik Anggraeni S.Si, M.Kom

## **ABSTRACT**

*The common problem in managing organization's supply chain is uncertainty of demand. Demand pattern is an important thing to note in order to support planning and managing that can increase supply chain effectively. Forecasting demand is a way to solve demand uncertainty, but not all forecasting methods can provide better levels of accuracy especially for irregular and intermittent demand patterns. Intermittent data pattern has multiple periods of zero demand, so that demand events occur only sporadically. It could be difficult to forecast because data with multiple periods of zero demand can't be solved using conventional time series methods such as moving average, exponential smoothing, holt winter, and linear regression. The results obtained by using these methods will result in a high error rate.*

*The method used in this final project is a Group Method of Data Handling (GMDH) network in which this method was assessed in previous studies to solving intermittent demand problems and providing better accuracy values when compared to other methods. GMDH is an artificial neural network structure which using polynomial transfer function and can build the model independently by making estimation based on Root Mean Square Error (RMSE) and iterating to get the best output.*

*The result shows that GMDH network model with the number of input neuron is 6, maximum number of neuron 15, number of hidden layer is 20, and value of selection pressure 0,2 is the best network structure model to forecast intermittent data. The RMSE value obtained is 14861 during training process and 21721 for the test data where the number is smaller than other models. In addition, MDA analysis shows that forecast generated from that model can follow the actual data pattern well. It is supported by value of MDA is obtain to 0.7983 for training and 0.7222 for testing*

**Keyword:** *Intermittent Data Forecasting, Intermittent Demand, Zero Demand, Group Method of Data Handling (GMDH)*

## KATA PENGANTAR

Puji Syukur kehadiran Tuhan Yang Maha Esa atas rahmat, hidayat, serta karunia-Nya yang diberikan sehingga penulis dapat menyelesaikan tugas akhir dengan judul:

### **“PENERAPAN METODE GROUP METHOD OF DATA HANDLING (GMDH) NETWORK PADA DATA INTERMITEN”.**

Dalam proses pelaksanaan dan penyusunan laporan tugas akhir ini, penulis banyak memperoleh ilmu, bantuan, serta saran dari berbagai pihak, sehingga penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT, karena atas berkat, rahmat dan karuni-Nya penulis dapat menyelesaikan kegiatan dan penyusunan laporan dengan tepat waktu.
2. Bapak Hartono dan Ibu Uswatun Khasanah sebagai orang tua yang senantiasa mendukung kegiatan tugas akhir dan mendoakan untuk kesuksesan tugas akhir.
3. Ibu Wiwik Anggraeni, S.Si, M.Kom selaku dosen pembimbing yang telah memberikan ilmu, nasihat, arahan pengerjaan, dan motivasi kepada penulis.
4. Bapak Edwin Riksakomara, S.Kom, M.T dan Bapak Faisal Mahananto, S.Kom, M.Eng, Ph.D selaku dosen penguji yang telah memberikan masukan untuk perbaikan pengerjaan tugas akhir.
5. Bapak Nisfu Asrul Sani selaku Kepala Program Studi S1 Sistem Informasi dan dosen wali yang telah memberikan arahan dan motivasi selama penulis menempuh perkuliahan.
6. Seluruh dosen Jurusan Sistem Informasi yang telah memberikan ilmu berharga kepada penulis.

7. Mas Ricky Asrul Sani selaku admin laboratorium yang telah memberikan tempat dan kenyamanan di laboratorium Rekayasa Data dan Itelegensi Bisnis.
8. Candra Anggadinata selaku teman yang memberikan motivasi, mendukung, dan membantu dalam penyelesaian tugas akhir.
9. Dina, Andina, dan Pramitya selaku teman yang membantu serta diskusi dalam penyelesaian tugas akhir.
10. Semua teman-teman Sistem Informasi Angkatan 2014 OSIRIS dan semua pihak yang mendukung serta memberikan semangat dalam pelaksanaan tugas akhir.

Penyusunan laporan tugas akhir ini masih jauh dari sempurna, untuk itu saya menerima adanya kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, Januari 2018

Penulis



## DAFTAR ISI

ABSTRAK.....	v
ABSTRACT .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR.....	xv
DAFTAR TABEL .....	xvii
DAFTAR SEGMENT KODE PROGRAM .....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Relevansi .....	4
1.7. Sistematika Penulisan .....	5
BAB II TINJAUAN PUSTAKA .....	7
2.1. Penelitian Sebelumnya .....	7
2.2. Dasar Teori.....	11
2.2.1. Peramalan.....	11
2.2.2. Data Intermiten.....	12
2.2.3. Transformasi Data Intermiten .....	12
2.2.4. Metode GMDH .....	13
2.2.6.1. Variabel Input.....	14
2.2.6.2. Konstruksi Model Parsial .....	15
2.2.6.3. Fungsi Transfer GMDH .....	16
2.2.6.4. Koefisien Fungsi Transfer .....	17
2.2.6.5. Seleksi Variabel pada Layer.....	18
2.2.6.6. Metode evaluasi model.....	20
2.2.7. Keunggulan Metode GMDH.....	22
BAB III METODOLOGI.....	25
3.2. Tahapan Pengerjaan Tugas Akhir .....	25
3.2.6. Identifikasi Masalah .....	26
3.2.7. Studi Literatur .....	26
3.2.8. Pengumpulan dan <i>Pra-Processing</i> Data .....	27
3.2.9. Menentukan Neuron Masukan .....	27

3.2.10.	Membuat Fungsi Transfer Polinomial .....	27
3.2.11.	Estimasi Koefisien Fungsi Transfer.....	28
3.2.12.	Menyeleksi Variabel Keluaran .....	28
3.2.13.	Mengecek Kriteria Penambahan Lapisan Berhenti	28
3.2.14.	Peramalan Data.....	29
3.2.15.	Analisa Hasil dan Penarikan Kesimpulan.....	29
3.2.16.	Penulisan Laporan Tugas Akhir .....	29
3.3.	Bahan dan Alat yang digunakan .....	30
3.3.6.	Bahan yang digunakan .....	30
3.3.7.	Alat yang digunakan.....	30
<b>BAB IV</b>	<b>PERANCANGAN .....</b>	<b>31</b>
4.1.	Pengumpulan dan <i>Pra-Processing</i> Data .....	31
4.1.1.	Pengumpulan Data.....	31
4.1.2.	<i>Pra-Processing</i> Data .....	32
4.2.	Penentuan Variabel dan Parameter .....	33
4.2.1.	Variabel Masukan (Input).....	33
4.2.2.	Variabel Keluaran pada Lapisan Tersembunyi.....	33
4.2.3.	Variabel Keluaran model jaringan GMDH .....	34
4.2.4.	Menentukan inisiasi parameter.....	34
4.3.	Perumusan Fungsi Transfer .....	35
4.4.	Estimasi koefisien Fungsi Transfer.....	36
4.5.	Menyeleksi Variabel Keluaran .....	36
4.6.	Mengecek Kriteria Penambahan Lapisan Berhenti.....	37
4.7.	Validasi Model GMDH .....	38
4.8.	Peramalan Data .....	38
<b>BAB V</b>	<b>IMPLEMENTASI.....</b>	<b>39</b>
5.1.	<i>Pra-processing</i> Data .....	39
5.2.	Menentukan Neuron Masukan.....	41
5.2.1.	Menentukan Lag Input .....	41
5.2.2.	Inisiasi Parameter .....	42
5.3.	Perumusan Fungsi Transfer .....	45
5.4.	Estimasi Koefisien Fungsi Transfer.....	47
5.5.	Menyeleksi Variabel Keluaran .....	49
5.6.	Mengecek Kriteria Penambahan Lapisan Berhenti.....	50
5.7.	Validasi Model GMDH .....	51
5.7.1.	Menghitung RMSE.....	54
5.7.2.	Menghitung MDA .....	54

5.7.3. Menghitung Correlation Factor (R).....	55
5.8. Plot Hasil Pemodelan .....	55
5.9. Peramalan Data.....	58
5.10. Plot Hasil Peramalan .....	59
BAB VI HASIL DAN PEMBAHASAN .....	60
6.1. Hasil Uji Coba .....	61
6.1.1. Lingkungan Uji Coba.....	61
6.1.2. Parameter dan Skenario Uji Coba .....	62
6.1.3. Hasil Uji Coba Model .....	63
6.1.4. Skenario Uji Coba Terbaik.....	67
6.2. Hasil Performa Model .....	67
6.2.1. Analisa RMSE dan MAD pada Model Jaringan GMDH.....	68
6.2.2. Analisa MDA pada Model Jaringan GMDH.....	72
6.2.3. Analisa <i>Correlation Factor</i> Output dan Target pada Model Jaringan GMDH.....	75
6.3. Struktur Jaringan Model Terbaik.....	78
6.4. Peramalan Periode Selanjutnya .....	80
BAB VII KESIMPULAN DAN SARAN .....	83
7.1. Kesimpulan.....	83
7.2. Saran.....	84
BIODATA PENULIS .....	88
LAMPIRAN A : Data Aktual Impor Beras Jawa Timur .....	A-1
LAMPIRAN B: Neuron Input .....	B-1
LAMPIRAN C : Variabel Target.....	C-1
LAMPIRAN D: Variabel Output Jaringan GMDH .....	D-1
LAMPIRAN E: Struktur Jaringan GMDH .....	E-1
LAMPIRAN F: Estimasi Koefisien Fungsi Transfer.....	F-1
LAMPIRAN G: Uji MDA .....	G-1
LAMPIRAN H: Uji Coba Model Berdasarkan Nilai RMSE..	H-1

*(Halaman ini sengaja dikosongkan)*

## DAFTAR GAMBAR

Gambar 2.1 Struktur Dasar GMDH .....	14
Gambar 2.2 Komputasi dengan Fungsi Transfer.....	16
Gambar 2.3 Arsitektur Jaringan GMDH dengan Neuron Terpilih.....	20
Gambar 3.1 Diagram Aliran Pengerjaan Tugas Akhir .....	25
Gambar 4.1 Data Realisasi Impor Beras 2002-2016.....	32
Gambar 4.2 Grafik Performa MSE Neuron Output .....	37
Gambar 5.1 Contoh Hasil Transformasi Data Aktual .....	39
Gambar 5.2 Contoh Perhitungan MDA.....	54
Gambar 6.1 Persebaran <i>Error</i> pada Data Pelatihan dan Pengujian Model ke-1 .....	69
Gambar 6.2 Persebaran <i>Error</i> pada Data Pelatihan dan Pengujian Model ke-2 .....	70
Gambar 6.3 Persebaran Error pada Data Pelatihan dan Pengujian Model ke-3 .....	71
Gambar 6.4 Perbandingan Pola Data Aktual dan Peramalan Model ke-1 .....	73
Gambar 6.5 Perbandingan Pola Data Aktual dan Peramalan Model ke-2 .....	74
Gambar 6.6 Perbandingan Pola Data Aktual dan Peramalan Model ke-3 .....	75
Gambar 6.7 Nilai Correlation Factor pada Data Pelatihan dan Pengujian Model ke-1 .....	76
Gambar 6.8 Nilai Correlation Factor pada Data Pelatihan dan Pengujian Model ke-2 .....	76
Gambar 6.9 Nilai Correlation Factor pada Data Pelatihan dan Pengujian Model ke-3 .....	77
Gambar 6.10 Struktur Jaringan GMDH Model Terbaik .....	78
Gambar 6.11 Performa Penurunan RMSE Neuron pada Setiap Lapisan .....	80
Gambar 6.12 Perbandingan Aktual dan Peramalan Impor Beras 2002-2018 .....	82
Gambar A 1 Data Aktual Impor Beras Bulog Jawa Timur 2002-2016 .....	A-1

Gambar B 1 Neuron Input Model ke-1 .....B-1

Gambar B 2 Neuron Input Model ke-2.....B-2

Gambar B 3 Neuron Input Model ke-3.....B-3

Gambar C 1 Variabel Target (Yt) Model ke-1 .....C-1

Gambar C 2 Variabel Tarhet (Yt) Model ke-2 .....C-2

Gambar C 3 Variabel Target (Yt) Model ke-3 .....C-3

Gambar D 1 Variabel Output (Ft) Jaringan GMDH Model ke-1  
..... D-1

Gambar D 2 Variabel Output (Ft) Jaringan GMDH Model ke-2  
..... D-2

Gambar D 3 Variabel Output (Ft) Jaringan GMDH Model ke-3  
..... D-3

Gambar E 1 Struktur Jaringan Model ke-1 .....E-1

Gambar E 2 Struktur Jaringan Model Terbaik ke-2 .....E-1

Gambar E 3 Struktur Jaringan Model Terbaik ke-3 .....E-1

## DAFTAR TABEL

Tabel 2.1 Ringkasan Jurnal Penelitian Sebelumnya .....	7
Tabel 6.1 Lingkungan Uji Coba.....	61
Tabel 6.2 Uji Coba Tanpa Menggunakan Transformasi Data.....	63
Tabel 6.3 Uji Coba Menggunakan Transformasi Data.....	64
Tabel 6.4 Uji Coba terhadap Penambahan Maksimum Lapisan .....	65
Tabel 6.5 Uji Coba terhadap Penambahan Nilai Alpha 0,1-0,5 .....	66
Tabel 6.6 Uji Coba terhadap Penambahan Nilai Alpha 0,6-1 .....	66
Tabel 6.7 Nilai RMSE Model Terbaik .....	67
Tabel 6.8 Tiga Model Terbaik.....	68
Tabel 6.9 Perbandingan Nilai RMSE.....	71
Tabel 6.10 Analisa Nilai Perbandingan MDA .....	72
Tabel 6.11 Pemilihan Model Terbaik Berdasarkan Nilai Korelasi Output dan Target .....	77
Tabel 6.12 Jumlah Neuron Terbentuk dan Penurunan RMSE Neuron pada struktur jaringan GMDH Model Terbaik..	79
Tabel 6.13 Hasil Peramalan 2 tahun Mendatang.....	81
Tabel G 1 Hasil Output dan Uji MDA terhadap Hasil peramalan Model ke-1 .....	G-1
Tabel G 2 Hasil Output dan Uji MDA terhadap Hasil peramalan Model ke-2.....	G-8
Tabel G 3 Hasil Output dan Uji MDA terhadap Hasil peramalan Model ke-3.....	G-15
Tabel H 1 Uji Coba Penambahan Maksimum Lapisan .....	H-1
Tabel H 2 Uji Coba Penambahan <i>Selection Pressure</i> 0,1-0,5 ....	H-3
Tabel H 3 Uji Coba Terhadap Penambahan <i>Selection Pressure</i> 0,6-1 .....	H-11





## **DAFTAR SEGMENT KODE PROGRAM**

Segmen Kode Program 5.1 Implementasi Transformasi Data	39
Segmen Kode Program 5.2 Pembagian Data Training dan Data Testing.....	40
Segmen Kode Program 5.3 Mendefinisikan Lag Input .....	41
Segmen Kode Program 5.4 Membuat Matriks Neuron Input	42
Segmen Kode Program 5.5 Inisiasi Parameter Jaringan GMDH .....	42
Segmen Kode Program 5.6 Fungsi Pelatihan Jaringan GMDH .....	43
Segmen Kode Program 5.7 Pembagian data training dan testing pada proses pelatihan jaringan.....	44
Segmen Kode Program 5.8 Variabel Pembentukan Lapisan Jaringan GMDH .....	45
Segmen Kode Program 5.9 Fungsi Pembentukan Lapisan Jaringan GMDH .....	46
Segmen Kode Program 5.10 Mengestimasi Koefisien Fungsi Transfer .....	47
Segmen Kode Program 5.11 Substitusi Estimasi Fungsi Transfer terhadap Data Testing .....	48
Segmen Kode Program 5.12 Perumusan Fungsi Transfer Menggunakan Polinomial Kuadrat Dua Variabel .....	49
Segmen Kode Program 5.13 Pembentukan Lapisan Jaringan GMDH dan Pengurutan Neuron.....	49
Segmen Kode Program 5.14 Perulangan Penambahan Lapisan pada Pelatihan Jaringan GMDH.....	51
Segmen Kode Program 5.15 Pembentukan Neuron Output Akhir pada Setiap Periode.....	52
Segmen Kode Program 5.16 Implementasi Model Jaringan GMDH.....	53
Segmen Kode Program 5.17 Perhitungan RMSE .....	54
Segmen Kode Program 5.18 Implementasi Perhitungan Correlation Factor .....	55
Segmen Kode Program 5.19 Plotting Target dan Output Data Training .....	56

Segmen Kode Program 5.20 Plotting Target dan Output Data Testing .....	56
Segmen Kode Program 5.21 Plotting Target dan Output Data Training dan Testing.....	56
Segmen Kode Program 5.22 Fungsi Plotting Data Training dan Testing pada Grafik Garis.....	57
Segmen Kode Program 5.23 Peramalan Data untuk Beberapa Periode ke Depan.....	58
Segmen Kode Program 5.24 Membuat Matriks Neuron Input Data Peramalan.....	58
Segmen Kode Program 5.25 Plotting Target dan Output Data Peramalan .....	59
Segmen Kode Program 5.26 Fungsi Plotting Data Peramalan pada Grafik Garis .....	59

## **BAB I**

### **PENDAHULUAN**

Pada bab pendahuluan akan diuraikan proses identifikasi masalah yang meliputi latar belakang masalah, perumusan masalah, batasan, tujuan, manfaat, relevansi dan sistematika penulisan laporan tugas akhir.

#### **1.1. Latar Belakang**

Manajemen logistik merupakan bagian terpenting dari proses manajemen rantai pasok [1]. Pada proses pengelolaan rantai pasok tersebut dilakukan koordinasi dan mengintegrasikan aktivitas-aktivitas yang melibatkan pengiriman produk mulai dari bahan mentah hingga tersampaikan kepada pelanggan akhir. Kendala umum yang banyak terjadi di perusahaan adalah ketidakpastian permintaan yang menyebabkan permasalahan persediaan seperti kelebihan atau kekurangan persediaan yang dapat menghambat proses bisnis perusahaan. Jika perusahaan ingin bersaing, maka perusahaan tersebut harus memiliki sistem manajemen logistik yang berkembang dengan baik di seluruh rantai pasokan [2]. Sub-sistem logistik utama dari sistem logistik umum sebuah perusahaan industri adalah pembelian, produksi, pergudangan, pengemasan, distribusi dan manajemen aliran pengembalian material dimana semua hal itu perlu didasarkan pada perkiraan permintaan yang tepat dan akurat. Pola permintaan baik teratur maupun tidak akan mendorong semua aktivitas logistik dalam rantai pasok, sehingga penting untuk dilakukan pengelolaan proses sistem logistik perusahaan yang efektif. Untuk mengatur hal tersebut pada proses pengelolaan logistik juga harus didasarkan pada perkiraan permintaan yang akurat. Oleh karena itu peramalan sangat diperlukan dalam mendukung perencanaan perusahaan dalam memenuhi permintaan pelanggan dan dapat diambil sebagai elemen penting yang perlu dilakukan pengembangan peramalan lebih lanjut. Hal ini dilakukan untuk dapat mendukung perencanaan sistem manajemen logistik perusahaan yang kompleks.

Pada kenyataannya terdapat banyak kasus dimana barang dalam sistem pengendalian persediaan jarang ada permintaan, sehingga permintaan bersifat sporadis atau terputus-putus. Hal tersebut disebut sebagai data intermiten dimana pola permintaan tersebut berbeda dengan pola permintaan deret waktu konvensional karena data intermiten memiliki banyak periode yang tidak terdapat permintaan atau permintaan nol [3]. Karena tidak beraturan dan sebagian besar nilainya adalah nol maka peramalan data tersebut menjadi sulit dan tidak pasti [4]. Mengelola ketidakpastian ini adalah masalah yang rumit dan sebagian besar belum dieksplorasi lebih lanjut. Contoh kasus umum yang memiliki pola data intermiten yaitu seperti permintaan mesin berat, suku cadang pesawat terbang, elektronik, suku cadang maritim, dan lain-lain. Peramalan data intermiten ini memainkan peran penting dalam manajemen rantai pasok dan manufaktur yang mengarah pada pemanfaatan kapasitas optimal dan optimalisasi biaya untuk bisnis, sehingga hal tersebut menjadi kebutuhan yang signifikan.

Peramalan data intermiten saat ini telah menjadi isu yang sangat menantang dalam literatur peramalan [5]. Banyak jurnal penelitian saat ini yang melakukan penilaian berbagai metode yang dapat diterapkan pada peramalan permintaan intermiten. Terdapat beberapa metode penyelesaian peramalan deret waktu konvensional seperti *moving average*, *exponential smoothing*, regresi linier tidak dapat menyelesaikan peramalan data intermiten karena tingkat akurasi yang rendah [6]. Salah satu peneliti seperti Croston juga menjawab masalah data intermiten dan menentukan metode baru yang menggunakan perkiraan ukuran dan frekuensi permintaan yang terpisah, dikenal dengan sebutan metode Croston [3]. Namun metode tersebut memiliki keterbatasan tersendiri pada kondisi tertentu tidak menghasilkan output yang akurat karena selalu ada bias yang meningkatkan nilai parameter  $\alpha$  [3]. Pengembangan metode lain seperti ARIMA, pendekatan jaringan syaraf tiruan (ANN) juga hasilnya kurang akurat jika dibandingkan dengan metode jaringan Group Method of Data Handling (GMDH) [7]. Rekomendasi berdasarkan penelitian tersebut adalah

menggunakan metode jaringan GMDH karena metode tersebut dapat menghasilkan keakuratan yang lebih baik pada pola data permintaan yang intermiten. Oleh karena itu, tugas akhir ini mengusulkan model jaringan GMDH sebagai metode untuk melakukan peramalan data intermiten. GMDH merupakan struktur jaringan yang menerapkan polinomial sebagai fungsi transfer yang mampu merumuskan model dengan *self-organizing* pada sistem yang kompleks dan non-linier dengan *penambahan lapisan* yang meminimalkan kesalahan prediksi pada setiap fase [5]. Dengan demikian, permasalahan data intermiten akan dapat diselesaikan dengan menggunakan metode GMDH, sehingga nantinya tugas akhir ini dapat dijadikan sebagai referensi untuk penyelesaian permasalahan semua kasus pola permintaan yang intermiten.

## **1.2. Rumusan Masalah**

Berdasarkan uraian latar belakang di atas, maka rumusan masalah yang akan diselesaikan pada tugas akhir ini adalah:

1. Bagaimana model GMDH terbaik yang dapat digunakan untuk meramalkan data intermiten
2. Bagaimana hasil dan tingkat akurasi peramalan data intermiten dengan menggunakan metode GMDH

## **1.3. Batasan Masalah**

Berdasarkan rumusan masalah yang disebutkan di atas, batasan masalah dari tugas akhir ini adalah:

1. Peramalan ini hanya diterapkan pada studi kasus impor beras di wilayah Jawa Timur
2. Data yang digunakan untuk peramalan adalah data periode bulanan sejak tahun 2002 hingga tahun 2016
3. Peramalan deret waktu yang diimplementasikan adalah peramalan impor beras di wilayah Jawa Timur periode bulanan untuk dua tahun selanjutnya yaitu tahun 2017-2018.
4. Pada tugas akhir ini menggunakan perangkat lunak MATLAB R2013a dengan *toolbox MATLAB exchange* fungsi GMDH yang dibuat oleh *Yarpiz Project*.

#### **1.4. Tujuan**

Berdasarkan penjelasan pada permasalahan di atas, maka tujuan dari tugas akhir ini adalah:

1. Untuk memperoleh model GMDH terbaik yang dapat digunakan untuk meramalkan data intermiten
2. Untuk mengetahui hasil dan tingkat akurasi peramalan data intermiten dengan menggunakan metode GMDH

#### **1.5. Manfaat**

Manfaat yang diharapkan dari penulisan tugas akhir ini:

##### **Bagi Penulis**

Mendapatkan pengetahuan serta dapat mempraktikkan atau mengimplementasikan dan mengembangkan ilmu pengetahuan serta metode yang diperoleh selama perkuliahan pada studi kasus nyata.

##### **Bagi Instansi**

Dapat memudahkan instansi atau organisasi dalam melakukan pengambilan keputusan terkait peramalan data intermiten. Dengan demikian dapat mendukung proses perencanaan dan pengelolaan rantai pasok.

##### **Bagi Masyarakat**

Menambah sumber pengetahuan dan pembelajaran bagi masyarakat terutama mengenai peramalan intermiten yang dapat mendukung keputusan. Pembelajaran tersebut nantinya dapat diterapkan dalam penyelesaian suatu permasalahan yang serupa serta dapat dijadikan acuan untuk mengembangkan penelitian selanjutnya.

#### **1.6. Relevansi**

Tugas akhir ini berkaitan dengan Laboratorium Rekayasa Data dan Inteligensi Bisnis (RDIB) pada bidang keilmuan *Computerized Decision Support*. Bidang keilmuan ini terbagi dalam dua cabang ilmu yaitu Sistem Pendukung Keputusan dan Sistem Pemodelan dan Analisis. Pada tugas akhir ini dipetakan ke dalam salah satu bidang keilmuan tersebut, yaitu Sistem Pemodelan dan Analisis. Mata kuliah yang terkait adalah

diantaranya: Statistika, Teknik Peramalan, dan Sistem Pendukung Keputusan.

Pada setiap rantai pasok organisasi atau perusahaan terdapat berbagai pola permintaan, baik teratur maupun tidak. Pola data intermiten merupakan pola permintaan yang tidak teratur dimana suatu permintaan terjadi secara terputus-putus [3]. Pola permintaan yang intermiten ini tidak dapat diselesaikan dengan metode deret waktu konvensional [6]. Ketidakteraturan pola data ini menimbulkan pengembangan metode tertentu untuk menangani data tersebut, sehingga tugas akhir ini akan menyelesaikan permasalahan data intermiten dengan menerapkan metode GMDH. Hal ini dilakukan karena pada studi literatur ditemukan banyak penelitian pada akhir-akhir ini yang menggunakan GMDH serta metode GMDH cukup dinilai baik dalam menyelesaikan permasalahan peramalan data intermiten [5]. Dengan adanya hasil peramalan yang mendekati data aktual maka hasil peramalan tersebut dapat digunakan sebagai pendukung perencanaan, pengelolaan, dan pendukung keputusan oleh instansi terkait.

### **1.7. Sistematika Penulisan**

Sistematika penulisan laporan tugas akhir disesuaikan dengan format yang telah ditentukan dan pengerjaan tugas akhir yang terdiri dari tujuh bab, yaitu sebagai berikut:

## **BAB I PENDAHULUAN**

Pada bab ini dijelaskan mengenai hal-hal penting yang menjadi latar belakang dari tugas akhir ini. Selain itu juga dijelaskan mengenai pentingnya dilakukan peramalan untuk data intermiten ini melalui rumusan masalah, tujuan, batasan, manfaat, serta relevansi tugas akhir.

## **BAB II TINJAUAN PUSTAKA**

Pada bab ini dijelaskan mengenai dasar teori metode GMDH dan penerapannya pada kasus-kasus tertentu. Selain dasar teori, juga dilakukan ringkasan jurnal penelitian sebelumnya yang memiliki keterkaitan dengan topik tugas akhir ini. Kedua hal

tersebut akan dijadikan sebagai referensi untuk pengerjaan tugas akhir ini.

### **BAB III METODOLOGI**

Pada bab ini dijelaskan mengenai alur sistematis melakukan peramalan dengan menggunakan metode GMDH. Alur sistematis ini akan digunakan sebagai tahapan pengerjaan tugas akhir. Metodologi atau tahapan yang diterapkan yaitu meliputi: identifikasi masalah, pengumpulan dan pra processing data, pemodelan, peramalan, analisa hasil peramalan, dan penyusunan laporan tugas akhir

### **BAB IV PERANCANGAN**

Pada bab ini akan dilakukan perancangan masukan dan keluaran data serta teknik *pra-processing* data yang akan digunakan

### **BAB V IMPLEMENTASI**

Pada bab ini akan dijelaskan mengenai proses pelaksanaan peramalan menggunakan metode GMDH yang sudah dirancang pada bab perancangan. Proses pelaksanaan juga akan dilengkapi dengan penjelasan *source code* dari *tools* matlab.

### **BAB VI HASIL DAN PEMBAHASAN**

Pada bab ini akan dilakukan analisa terhadap hasil peramalan yang sudah dilaksanakan pada bab implentasi. Analisa ini dilakukan berdasarkan RMSE (*Root Mean Square Error*) untuk dapat mengetahui tingkat akurasi peramalan.

### **BAB VII KESIMPULAN DAN SARAN**

Setelah menganalisa hasil peramalan berdasarkan tingkat akurasi, maka dapat dilakukan penarikan kesimpulan dari pengerjaan tugas akhir ini serta menuliskan saran – saran yang tepat yang dapat dikembangkan untuk penelitian selanjutnya.



## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini dijelaskan mengenai penelitian sebelumnya dan dasar teori yang akan dijadikan sebagai acuan untuk pengerjaan tugas akhir. Dasar teori merupakan gambaran umum tentang studi kasus dan penjelasan teori mengenai metode yang akan diterapkan pada studi kasus. Penelitian sebelumnya merupakan penelitian yang memiliki keterkaitan dengan pengerjaan tugas akhir ini.

#### **2.1. Penelitian Sebelumnya**

Dalam melakukan studi literatur untuk pengerjaan tugas akhir, didapatkan beberapa jurnal penelitian sebelumnya dan penelitian tersebut memiliki keterkaitan atau berhubungan dengan tugas akhir ini. Pada tugas akhir ini diperlukan jurnal penelitian sebelumnya yang menjelaskan tentang permintaan intermiten serta beberapa metode-metode yang tepat digunakan untuk menangani permasalahan tersebut. Tabel 2.1 merupakan beberapa penelitian yang dapat dijadikan sebagai acuan untuk pengerjaan tugas akhir:

**Tabel 2.1 Ringkasan Jurnal Penelitian Sebelumnya**

Penelitian 1 [5]	
Judul Penelitian	<i>Intermittent Demand Forecast: Robustness Assessment for Group Method of Data Handling</i>
Penulis/ Tahun Penelitian	Prerna Mishra, Xue-Ming Yuan, Guangbin Huang, Truong Ton Hien Due/2014
Gambaran Umum	Pada penelitian ini melakukan peramalan terhadap intermiten demand pada data 1500 produk manufaktur. Kemudian dilakukan perbandingan metode peramalan moving average, Simple Exponential Smoothing, Croston, dan GMDH. Hasil

	yang ditemukan adalah metode GMDH dapat memberikan tingkat akurasi yang lebih baik dengan memperhatikan nilai MSE (Mean Square Error)
Keterkaitan Tugas Akhir	Referensi dalam memilih GMDH sebagai metode peramalan yang tepat untuk menyelesaikan permasalahan data intermiten
Penelitian 2 [6]	
Judul Penelitian	<i>Intermittent Demand with Neural Network</i>
Penulis/ Tahun Penelitian	Nikolaos Kourentzes/2013
Gambaran Umum	Pada jurnal penelitian ini menerapkan metode neural network untuk menangani data intermiten dengan membandingkan metode Croston, SES, dan moving average. Hasil yang diperoleh menunjukkan bahwa neural network lebih efektif untuk meramalkan data yang terjadi secara sporadic dengan menggunakan matriks akurasi peramalan untuk mengevaluasi hasil peramalan menggunakan beberapa metode tersebut.
Keterkaitan Tugas Akhir	Menunjukkan beberapa metode yang dapat digunakan untuk menyelesaikan permasalahan pada peramalan data intermiten.

Penelitian 3 [7]	
Judul Penelitian	<i>A Hybrid GMDH and Box-Jenkins Model in Time Series Forecasting</i>
Penulis/ Tahun Penelitian	Ani Shabri, Ruhaidah Samsudin / 2014
Gambaran Umum	Pada jurnal penelitian ini dilakukan perbandingan beberapa metode untuk peramalan time series. Metode tersebut diantaranya: ARIMA, ANN, dan GMDH. Hasil yang diperoleh yaitu metode GMDH menghasilkan akurasi yang lebih baik dengan memperhatikan nilai MSE dari semua model ketiga metode tersebut.
Keterkaitan Tugas Akhir	Dapat dijadikan referensi mengenai penelitian yang pernah dilakukan dengan menggunakan metode GMDH secara detail dengan step step pengerjaan dalam studi kasus.
Penelitian 4 [8]	
Judul Penelitian	<i>Application of GMDH to Short-Term Load Forecasting</i>
Penulis/ Tahun Penelitian	Hongya Xu, Yao Dong, Jie Wu, Weigang Zhao/2012
Gambaran Umum	Pada jurnal penelitian ini melakukan peramalan time series untuk beban daya harian. Peramalan beban daya harian memainkan peran penting dalam operasi dan perencanaan sistem tenaga listrik. Oleh karena itu, perlu dicari keterkaitan data secara otomatis dan pilih struktur

	model yang optimal. <i>Group Method of Data Handling (GMDH)</i> diterapkan untuk meramalkan permintaan beban listrik New South Wales (NSW) di Australia mulai 17 Januari 2009 sampai 18 Januari 2009. Dibandingkan dengan hasil yang diperoleh oleh ARIMA, kami menunjukkan bahwa GMDH adalah metode yang lebih baik untuk peramalan time series jangka pendek.
Keterkaitan Tugas Akhir	Dapat dijadikan referensi mengenai penelitian yang pernah dilakukan dengan menggunakan metode GMDH dalam mengerjakan tugas akhir
Penelitian 5 [9]	
Judul Penelitian	<i>Improved Holt-Winters Method: A Case Overnight Stays of Tourist in Republic of Slovenia</i>
Penulis/ Tahun Penelitian	Liljana Ferbar Tratar /2013
Gambaran Umum	Pada jurnal penelitian ini dilakukan peramalan terhadap data penginapan wisatawan atau pengunjung di Slovenia dengan menggunakan pengembangan metode <i>additive Holt-Winter</i> . Alasan menggunakan metode tersebut karena metode <i>Holt-Winter</i> konvensional tidak cukup akurat untuk memprediksikan data yang pada waktu tertentu terdapat nilai nol atau tidak ada permintaan penginapan di Slovenia. Hasil peramalan dapat diterima

	karena menghasilkan nilai MSE yang rendah.
Keterkaitan Tugas Akhir	Menunjukkan beberapa metode yang dapat digunakan untuk menyelesaikan permasalahan pada peramalan data intermiten

## 2.2. Dasar Teori

Pada sub bab ini dijelaskan mengenai dasar teori peramalan secara umum, pola permintaan yang intermiten serta dasar teori metode yang dapat menyelesaikan permasalahan pola data intermiten

### 2.2.1. Peramalan

Peralaman dapat memberikan kemungkinan yang obyektif untuk mendukung perencanaan dan keputusan bisnis. Peramalan tidak sama dengan perencanaan, namun peralaman digunakan dalam proses pembuatan keputusan yang berguna untuk meningkatkan akurasi pada pembuat keputusan [10]. Tujuan dari sistem peramalan dapat memberikan informasi tentang perubahan masa depan di lingkungan bisnis dan dampak dari perubahan dalam bentuk prakiraan, sehingga dengan adanya peramalan tersebut dapat digunakan sebagai masukan untuk melakukan perencanaan dalam bisnis atau organisasi.

Sebagian besar peramalan melibatkan penggunaan data deret waktu (*time series*). *Time series* merupakan urutan atau kronologis pada variabel yang diamati [11]. Tujuan utama pemodelan *time series* adalah untuk mengumpulkan dan mempelajari secara seksama mengenai pengamatan terakhir dari deret waktu untuk mengembangkan model yang sesuai atau dapat disebut sebagai tindakan meramalkan masa depan secara kuantitatif dengan memperhatikan data masa lalu.

### 2.2.2. Data Intermiten

Data intermiten merupakan suatu pola data permintaan yang sangat random dengan terdapat banyak nilai kosong atau tidak ada permintaan pada beberapa periode [4]. Menurut Croston, data atau permintaan intermiten adalah permintaan yang sporadis atau terputus-putus [3]. Terdapat banyak kasus atau kejadian di mana barang dalam sistem pengendalian persediaan mengalami permintaan yang jarang. *Time series* yang mengalami data intermiten berbeda dengan *time series* konvensional karena data intermiten memiliki banyak periode dengan jumlah permintaan nol. Permintaan yang intermiten pada umumnya terjadi pada barang atau modal dengan harga tinggi misalnya seperti suku cadang dan alat berat [6]. Peramalan data intermiten memainkan peran penting dalam manajemen rantai pasok dan manufaktur yang mengarah pada pemanfaatan kapasitas optimal, optimalisasi biaya dan layanan yang tepat untuk bisnis. Dalam melakukan pemodelan pola data intermiten diperlukan identifikasi dan penerapan alat pemodelan yang tepat. Beragam metodologi yang menggunakan *time series* seperti *moving average*, dan *exponential smoothing* telah dipublikasikan secara luas untuk mencapai perkiraan data intermiten dengan menggunakan informasi historis permintaan dengan beberapa permintaan nol pada periode tertentu.

### 2.2.3. Transformasi Data Intermiten

Suatu kumpulan data perlu dilakukan transformasi untuk memperbaiki asumsi normalitas, linearitas, homogenitas varians. Transformasi log dapat mengompres nilai yang tinggi dan menyebarkan nilai rendah dengan merepresentasikan nilai sebagai urutan besarnya [12]. Transformasi log sering digunakan apabila terdapat variasi tingkat tinggi dalam variabel atau bila ada variasi tingkat tinggi di antara atribut dalam sampel.

Untuk transformasi logaritma dengan menggunakan data yang berisi angka nol, maka harus ditambahkan koefisien

yang merupakan nilai minimum pada data yang selain nilai nol. Formula yang digunakan untuk transformasi adalah [12]:

$$Y_{trans} = \log(Y_t + c) \quad (2.2.1)$$

dimana

$Y_{trans}$  = data hasil transformasi

$Y_t$  = data aktual

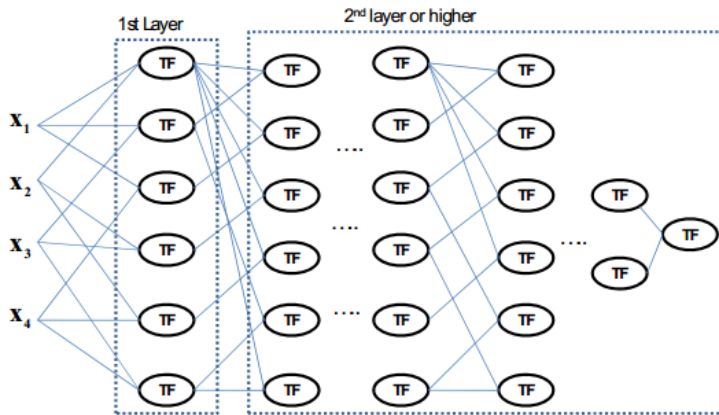
C = konstanta yang merupakan nilai minimum selain nol dari data aktual

#### 2.2.4. Metode GMDH

GMDH (*Group Method of Data Handling*) adalah algoritma yang ditemukan oleh Ivakhenho yang memungkinkan pencapaian model matematis sistem dari database dan dapat digunakan dalam pemilihan variabel input otomatis [13]. Gagasan utama GMDH adalah membangun fungsi analitik dalam jaringan *feedforward* berdasarkan fungsi transfer yang koefisiennya diestimasi dengan menggunakan RMSE (*Root Mean Square Error*). Jaringan GMDH memiliki struktur berlapis-lapis tertentu yang ditentukan dari proses pelatihan dan dapat digunakan untuk mendeteksi hubungan input-output untuk berbagai model [14]. Hal ini bertujuan untuk menemukan hubungan antara satu keluaran dan serangkaian kemungkinan masukan yang sering terjadi. Jaringan akan mengidentifikasi dan memutuskan mana dari input yang benar-benar relevan dengan hasil keluaran yang diharapkan. Oleh karena itu, jaringan dibangun secara berlapis-lapis selama pelatihan. Di setiap lapisan, terdapat beberapa neuron dimana setiap neuron hanya memiliki masing-masing dua masukan. Output dari setiap neuron itu merupakan fungsi kuadrat dari kedua inputnya. Parameter fungsi kuadrat diperoleh dengan menggunakan analisis regresi [15]. Sebelum menambahkan lapisan baru, lapisan sebelumnya dilatih menggunakan data pelatihan. Neuron akan dilatih untuk setiap kombinasi unik dari dua masukan berdasarkan kriteria seleksi

tertentu dan hanya akan ada neuron berperforma terbaik yang dipilih. Kemudian lapisan baru ditambahkan dan seluruh prosedur latihan dilakukan lagi pada layer baru ini. Perulangan untuk penambahan lapisan jaringan akan terus dilakukan selama terdapat perbaikan kesalahan dari lapisan sebelumnya ke lapisan saat ini.

GMDH juga merupakan metode pemodelan *self-organizing* yang dihasilkan secara adaptif dari data dalam bentuk jaringan neuron aktif dan melakukan generasi berulang dari populasi model, menerapkan validasi yang sesuai, serta menerapkan seleksi model sampai terbentuk model yang optimal.



**Gambar 2.1 Struktur Dasar GMDH**

Gambar 2.1 merupakan struktur dasar GMDH yang terdiri dari titik masukan (*neuron input*), lapisan tersembunyi (*hidden layer*), dan hasil keluaran (*output*) [15].

#### 2.2.6.1. Variabel Input

Secara umum, data atau variabel yang menjadi masukan dan keluaran pada sistem direpresentasikan dengan input variabel matrik  $X = \{x_{ij}\}$ , dan variabel keluaran yaitu vektor  $Y = y_i$ ,  $i = 1, 2, 3, \dots, n$  dan  $j = 1, 2, 3, \dots, m$  dimana  $n$  merupakan banyaknya pengamatan dan  $m$  merupakan banyaknya variabel



input. Persamaan 2.2.2 merupakan input vektor sebagai lapisan pertama pada GMDH [16].

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & x_{ij} & \dots \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_i \\ \dots \\ y_n \end{bmatrix} \quad (2.2.2)$$

Pada peramalan time series, variabel input yang digunakan yaitu beberapa periode sebelumnya yang direpresentasikan sebagai persamaan 2.2.3 [7]:

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}) \quad (2.2.3)$$

Namun sampai saat ini tidak terdapat cara sistematis untuk menentukan nilai  $p$  yang optimum [15].

#### 2.2.6.2. Konstruksi Model Parsial

Jaringan GMDH adalah teknik pemodelan berbasis data yang menggunakan fungsi matematis untuk mengkarakterisasi hubungan nonlinear yang kompleks di antara masukan data keluaran yang diberikan. Jaringan GMDH terdiri dari sejumlah lapisan yang mengandung neuron. Setiap neuron memiliki dua *input* dan satu *output* tunggal. Keluaran dari setiap neuron dihitung dengan menggunakan fungsi transfer polinomial Ivakhnenko yang dijelaskan pada Persamaan 2.2.4 [17]:

$$y = a_0 + \sum_{i=1}^M a_i x_i + \sum_{i=1}^M \sum_{j=1}^M a_{ij} x_i x_j + \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M a_{ijk} x_i x_j x_k + \dots \quad (2.2.4)$$

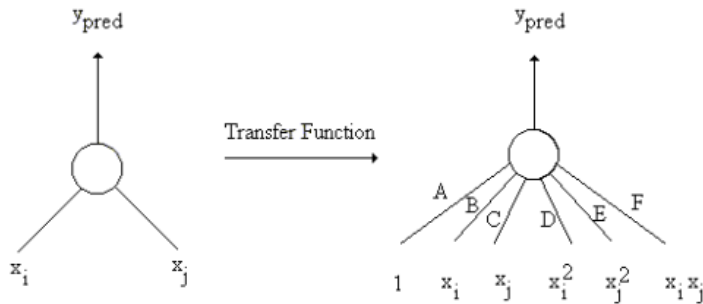
dimana  $(x_i, x_j, x_k, \dots)$  merupakan input vektor variabel,  $M$  merupakan jumlah input, dan  $(a_0, a_i, a_{ij}, a_{ijk}, \dots)$  vektor dari jumlah koefisien. Pada penerapan umumnya menggunakan dua variabel yang digunakan pada persamaan tersebut dan disebut sebagai polinomial kuadrat parsial. Untuk membentuk inisiasi kombinasi polinomial kuadrat parsial dari 2 masukan, sehingga nantinya jumlah neuron pada setiap *hidden layer* akan memiliki jumlah maksimum sebanyak jumlah kombinasi neuron *input*. Sehingga untuk menghitung maksimum neuron pada lapisan selanjutnya, maka dapat menggunakan persamaan 2.2.5 [7]:

$$L = \frac{M(M-1)}{2} \quad (2.2.5)$$

dimana  $L$  merupakan inisiasi jumlah kombinasi input dan  $M$  merupakan jumlah neuron input pada lapisan pertama. Nilai  $L$  yang didapatkan akan digunakan untuk menentukan banyaknya neuron *output* untuk lapisan berikutnya

### 2.2.6.3. Fungsi Transfer GMDH

Rincian algoritma GMDH dan prosedur sistematis untuk memprediksi  $y$  sebagai fungsi dari  $\{x_1, x_2, x_3, \dots, x_m\}$  diberikan seperti pada Gambar 2.2 [16]:



**Gambar 2.2** Komputasi dengan Fungsi Transfer

Gambar 2.2 menunjukkan simpul komputasi setiap layer GMDH yang dibentuk dari fungsi transfer polinomial. Langkah pertama adalah mengambil semua variabel independen (kolom X) dua per satu untuk semua kombinasi yang mungkin dan membuat polinomial kuadrat parsial sehingga fungsi transfer parsial yang dibentuk adalah seperti pada Persamaan 2.2.6 [7]:

$$\begin{aligned} z_l = G(x_i, x_j) = & a_0 + a_1x_i + a_2x_j \quad (2.2.6) \\ & + a_3x_ix_j + a_4x_i^2 \\ & + a_5x_j^2 \end{aligned}$$

untuk  $l = 1, 2, 3, \dots, L$ ; dimana  $z_l$  merupakan neuron output pada lapisan tersembunyi pada kombinasi ke- $L$ ;  $a_1, a_2, a_3, a_4, a_5$  merupakan koefisien fungsi transfer polinomial; dan  $x_i, x_j$  merupakan input untuk setiap kombinasi.

#### 2.2.6.4. Koefisien Fungsi Transfer

Untuk membangun jaringan GMDH yang lengkap, kombinasi masing-masing dua variabel masukan di setiap lapisan dihasilkan. Kemudian untuk setiap kombinasi dan keluaran yang terkait, koefisien polinomial diestimasi dengan algoritma *least square error* menggunakan teknik regresi non-linear polinomial [7]. Data yang digunakan untuk mengestimasi koefisien fungsi transfer adalah data pelatihan. Setelah koefisien polinomial diperoleh, output dari polinomial kemudian dievaluasi dan diuji dengan menggunakan *Root Mean Square Error* (RMSE).

Koefisien dari fungsi transfer ditentukan berdasarkan *Root Mean Square Error* (RMSE) menggunakan Persamaan 2.2.7 [15]:

$$A_i = (X_i^T X_i)^{-1} X_i Y \quad (2.2.7)$$

dimana  $A = \{a_0, a_1, \dots, a_5\}$  merupakan vektor koefisien yang belum diketahui,  $Y = \{Y_1, Y_2, \dots, Y_n\}^T$  merupakan vektor nilai *output* dari observasi, dan  $X$  adalah

$$X = \begin{bmatrix} 1 & x_{1i} & x_{1j} & x_{1i}x_{1j} & x_{1i}^2 & x_{1j}^2 \\ 1 & x_{2i} & x_{2j} & x_{2i}x_{2j} & x_{2i}^2 & x_{2j}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{ni} & x_{nj} & x_{ni}x_{nj} & x_{ni}^2 & x_{nj}^2 \end{bmatrix} \quad (2.2.8)$$

#### 2.2.6.5. Seleksi Variabel pada Layer

Setiap lapisan yang dibentuk pada jaringan GMDH terdapat neuron yang akan dihitung nilai kesalahan pada model antara masukan aktual dan hasil keluaran setiap neuron. Kriteria kesalahan atau *error* yang dapat diterima juga menunjukkan kombinasi input mana yang lebih relevan dengan jaringan dan dapat menguji kemampuan neuron polinomial terhadap keluaran sistem yang diinginkan. Dalam melakukan seleksi variabel output, data yang digunakan adalah data pengujian. Semakin kecil kriteria kesalahan, semakin baik polinom neuron pada data. Kriteria kesalahan atau error dapat dinyatakan dengan Persamaan 2.2.9 [7]:

$$\text{RMSE} = \sqrt{\frac{1}{n^T} \sum_{i=1}^{n^T} (y_i - z_{i,k})^2}, \text{ untuk } k = 1, 2, 3, \dots, L \quad (2.2.9)$$

dimana RMSE merupakan kriteria kesalahan *Root Mean Square Error*,  $n^T$  menunjukkan jumlah data pengujian,  $y_i$  merupakan keluaran yang diharapkan pada periode ke- $i$  dan  $z_{i,k}$  merupakan output pada periode ke- $i$  dan pada neuron ke- $k$ .

Dalam jaringan GMDH, kriteria untuk setiap keluaran neuron dievaluasi. Kemudian, digunakan untuk menentukan neuron yang masih dilanjutkan ke lapisan berikutnya sesuai dengan prosedur sortir yang telah ditetapkan. Neuron berperforma terbaik di setiap lapisan GMDH dipilih berdasarkan RMSE yang terendah, dan neuron yang kurang dari nilai ambang batas RMSE yang telah ditentukan maka akan dipilih untuk menjadi input ke layer berikutnya, sedangkan neuron lainnya akan dihilangkan dan dibuang dari jaringan [14]. Ambang batas atau *threshold* harus ditetapkan yang tepat untuk menentukan jumlah neuron di setiap lapisan. Ambang batas ini akan mempengaruhi hasil akhir dari jaringan GMDH. Nilai seleksi yang digunakan yaitu  $(0 \leq \alpha \leq 1)$ . Nilai minimum dari seleksi yaitu 0 atau tidak ada tekanan dan nilai 1 untuk maksimum tekanan yang diberikan. Beberapa hasil penelitian menunjukkan bahwa dari banyak uji coba, telah dihasilkan model dengan performa RMSE terbaik yaitu dengan menginisiasi *selection pressure* ( $\alpha$ ) dengan nilai 0,6 – 0,7 [18]. Nilai *selection pressure* dikalkulasi sebagai *threshold* dan dibandingkan dengan nilai RMSE setiap neuron pada setiap lapisan. Persamaan 2.2.10 merupakan perhitungan *threshold* berdasarkan nilai *selection pressure* ( $\alpha$ ) yang diinputkan [18]:

$$\text{Threshold} = (\alpha \cdot \text{RMSE terendah neuron}) + ((1-\alpha) \cdot \text{RMSE tertinggi neuron}) \quad (2.2.10)$$

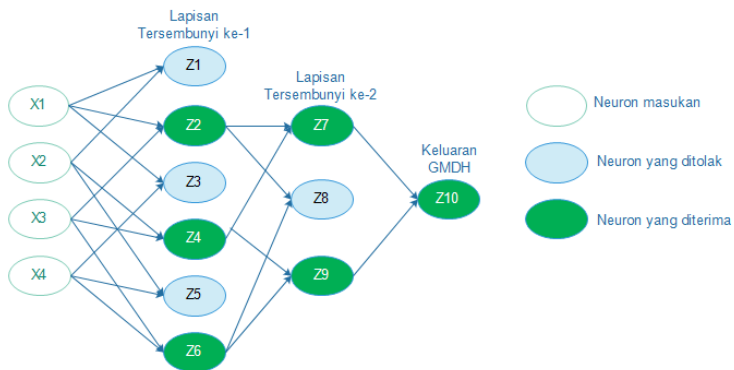
dimana *threshold* merupakan nilai ambang batas yang akan dibandingkan dengan RMSE neuron dan  $\alpha$  adalah nilai *selection pressure*

Jika neuron yang lolos terhadap ambang batas sangat banyak, maka jaringan GMDH menggunakan inisiasi maksimum neuron dan maksimum lapisan

untuk tetap dapat menghindari kompleksitas jaringan GMDH. Nilai maksimum neuron menggunakan kombinasi *lag input* seperti yang telah dijelaskan pada Persamaan 2.2.5 dan maksimum layer yang ditentukan. Penelitian sebelumnya menunjukkan bahwa dari banyak uji coba, telah dihasilkan

model dengan performa RMSE terbaik yaitu dengan menginisiasi maksimum layer dengan nilai 10 dan meningkatkan hingga 50 lapisan [18].

Neuron yang menghasilkan hasil paling buruk dan melebihi kriteria seleksi yang dipilih sebelumnya, maka harus dikeluarkan dari lapisan. Jika RMSE terkecil di lapisan berikutnya lebih besar dari RMSE terkecil di lapisan sebelumnya, maka penambahan layer akan berhenti [19]. Hasil akhir dari jaringan GMDH akan menjadi keluaran neuron dengan MSE terkecil pada lapisan terakhir seperti yang telah digambarkan pada Gambar 2.3 [14].



**Gambar 2.3 Arsitektur Jaringan GMDH dengan Neuron Terpilih**

Arsitektur yang terdapat pada Gambar 2.3 tersebut dapat dilihat bahwa setelah dilakukan proses seleksi neuron pada lapisan tersembunyi maka dapat diketahui variabel masukan apa saja yang relevan terhadap keluaran pada jaringan GMDH sehingga variabel tersebut dapat digunakan untuk proses peramalan.

#### 2.2.6.6. Metode evaluasi model

Evaluasi model diperlukan untuk mengukur kemampuan model yang ditemukan. Ada dua cara yang digunakan dalam

penelitian ini untuk mengukur keakurasian model yaitu RMSE dan MDA.

a. RMSE

RMSE (*Root Mean Square Error*) merupakan pengukuran *error* yang biasa digunakan dalam peramalan yang biasa disajikan dalam bentuk akar dari rata-rata kuadrat *error* [20]. Semakin kecil nilai RMSE yang dihasilkan, maka akan semakin bagus model yang dihasilkan. Persamaan 2.2.11 merupakan rumus perhitungan RMSE:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_t - g_t)^2}{n}} \quad (2.2.11)$$

dimana:

$n$  = jumlah data

$y_t$  = nilai aktual

$g_t$  = nilai output GMDH (hasil peramalan)

b. MDA

*Mean Directional Accuracy* (MDA) merupakan alat ukur akurasi yang digunakan untuk membandingkan arah data peramalan dan arah data aktual (naik dan turunnya), dengan kata lain bagaimana ketepatan model peramalan dalam mengikuti pola data aktual [21]. Persamaan 2.2.12 adalah persamaan untuk menentukan MDA.

$$\text{MDA} = \sum_t 1_{\text{sign}(A_t - A_{t-1}) = \text{sign}(F_t - F_{t-1})} \quad (2.2.12)$$

dimana:

$A$  = nilai aktual

$F$  = nilai peramalan

c. *Correlation Factor (R)*

Koefisien korelasi (R) adalah ukuran yang menggambarkan tingkat hubungan statistik antara dua variabel tingkat interval atau rasio. Koefisien korelasi diskalakan sehingga selalu antara -1 dan +1. Bila R mendekati 0 berarti bahwa ada sedikit hubungan antara variabel dan semakin mendekati angka 1 berarti terdapat hubungan yang signifikan antara kedua variabel [22]. Persamaan 2.2.13 adalah persamaan untuk menghitung nilai korelasi:

$$R = \frac{\sum_{i=1}^m (Y_{i(t)} - \bar{Y}_{(t)}) (Y_{i(out)} - \bar{Y}_{(out)})}{\sqrt{\sum_{i=1}^m (Y_{i(t)} - \bar{Y}_{(t)})^2 (Y_{i(out)} - \bar{Y}_{(out)})^2}} \quad (2.2.13)$$

dimana

$Y_{(t)}$  = nilai target (data aktual)

$Y_{(out)}$  = data peramalan (data peramalan)

### 2.2.7. Keunggulan Metode GMDH

GMDH merupakan algoritma yang dapat melakukan pengorganisasian diri secara induktif dengan model aljabar karena tidak perlu mengetahui model fisik pastinya terlebih dahulu [14]. Namun GMDH secara otomatis mempelajari hubungan yang mendominasi variabel sistem selama proses pelatihan. Dengan kata lain, struktur neuron yang optimal dipilih secara otomatis dengan cara meminimalkan nilai kesalahan prediksi dan neuron yang tidak perlu akan dihilangkan dari jaringan. Oleh karena itu, GMDH memiliki kemampuan generalisasi yang baik dan dapat menyesuaikan kompleksitas sistem non linier.

Algoritma GMDH dapat disebut sebagai jaringan syaraf polinomial, dimana fungsi pemrosesan neuron adalah fungsi polinomial bukan menggunakan fungsi sigmoid. Oleh karena itu, optimalisasi di GMDH didasarkan pada serangkaian pemasangan kuadrat-terkecil dan bukan merupakan metode



iteratif untuk meminimalkan kesalahan seperti jaringan saraf *back-propagation*. Keuntungan lain dari GMDH adalah tidak bisa *over-train* [14]. *Overtraining* jaringan syaraf bisa menjadi masalah bila tidak cukup banyak poin yang diberikan dan terlalu banyak perulangan penambahan lapisan yang digunakan untuk melatih jaringan. Algoritma pelatihan model GMDH, pelatihan berhenti saat mencapai konfigurasi terbaik. Selanjutnya, model akhir GMDH dinyatakan secara matematis dalam hal polinomial orde tinggi. Namun, model akhir yang diekstraksi oleh jaringan syaraf tiruan masih tersembunyi dan didistribusikan melalui jaringan. Selain itu, algoritma GMDH memiliki pendekatan sistematis, berlawanan dengan model jaringan syaraf tiruan yang didasarkan pada uji coba dan kesalahan. Singkatnya, perbedaan utama antara GMDH dan jaringan syaraf tiruan adalah kemampuan GMDH yang secara obyektif memilih model yang optimal, menghindari masalah *overfitting*, dan memilih variabel masukan yang paling relevan.

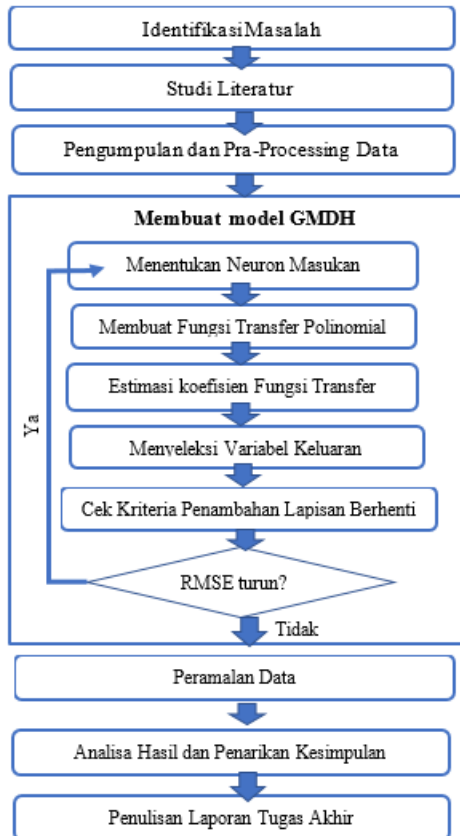
*(Halaman ini sengaja dikosongkan)*

### BAB III METODOLOGI

Pada bagian metodologi akan dijelaskan mengenai langkah-langkah sistematis sebagai panduan untuk mengerjakan penugasan tugas akhir.

#### 3.2. Tahapan Pengerjaan Tugas Akhir

Pada sub bab ini dijelaskan mengenai metodologi atau tahapan pengerjaan tugas akhir, meliputi identifikasi masalah hingga analisa hasil dan penarikan kesimpulan. Proses yang akan dilakukan ditunjukkan pada Gambar 3.1:



Gambar 3.1 Diagram Aliran Pengerjaan Tugas Akhir

### 3.2.6. Identifikasi Masalah

Tahap awal pengerjaan tugas akhir adalah melakukan identifikasi masalah. Identifikasi masalah dilakukan dengan mencari permasalahan umum pada pengelolaan rantai pasok perusahaan melalui studi literatur. Keluaran yang dihasilkan pada tahapan ini yaitu permasalahan yang diangkat pada tugas akhir ini. Banyak literatur yang menunjukkan bahwa permasalahan yang terjadi pada pengelolaan rantai pasok adalah ketidakpastian permintaan. Dalam menanggapi ketidakpastian permintaan dapat dilakukan peramalan, namun tidak semua metode peramalan dapat menyelesaikan semua jenis pola permintaan. Salah satu masalah pada pola permintaan yaitu pola data intermiten dimana terdapat beberapa periode yang tidak ada permintaan. Hal ini menyebabkan dibutuhkan metode tertentu untuk menyelesaikan masalah tersebut. Oleh karena itu dari permasalahan tersebut dapat diangkat sebagai topik tugas akhir.

### 3.2.7. Studi Literatur

Setelah melakukan identifikasi masalah, selanjutnya adalah studi literatur. Masukan pada tahapan ini yaitu referensi melalui buku, jurnal penelitian, tugas akhir sebelumnya, serta referensi terkait lainnya. Hal ini dilakukan untuk mendapatkan keluaran berupa solusi yang tepat berupa dasar teori dan penelitian pendukung yang dapat diterapkan pada permasalahan. Dasar teori yang didapatkan menjelaskan mengenai, pola data data intermiten, serta metode GMDH yang digunakan untuk menyelesaikan permasalahan peramalan data intermiten. Dengan adanya penelitian sebelumnya diharapkan dapat menemukan *gap* atau batasan yang dapat dikembangkan pada penulisan tugas akhir ini. Penelitian sebelumnya yang dijadikan studi literatur adalah yang memiliki keterkaitan dengan penerapan metode GMDH pada pola permintaan yang terjadi secara sporadic atau jarang, serta ada keterkaitan mengenai cara penyelesaian permasalahan data intermiten menggunakan beberapa metode.

### **3.2.8. Pengumpulan dan *Pra-Processing* Data**

Pada tahapan ini dilakukan pengumpulan data dari instansi terkait yang akan digunakan sebagai objek pada tugas akhir ini. Masukan pada tahapan ini yaitu data mentah realisasi impor beras dari Bulog Jawa Timur. Data yang digunakan untuk peramalan yaitu data realisasi impor beras periode bulanan pada tahun 2002 hingga 2016. Sebelum melakukan peramalan, maka perlu tahapan transformasi data untuk menghilangkan varian data yang sangat besar. Hal ini telah dijelaskan pada sub bab 2.2.3. Kemudian data dibagi menjadi dua bagian yang akan digunakan sebagai data *training* dan data *testing* selama proses peramalan. Keluaran pada tahapan ini adalah berupa hasil transformasi data dan pembagian data *training testing* sesuai dengan presentase atau proporsi yang ditentukan.

### **3.2.9. Menentukan Neuron Masukan**

Tahap awal untuk membuat arsitektur model GMDH adalah menentukan neuron masukan pada lapisan pertama. Masukan pada tahapan ini adalah data realisasi impor beras sedangkan keluaran dari tahapan ini adalah berupa inisiasi variabel input yang akan digunakan untuk pembentukan model. Untuk peramalan *time series* variabel input menggunakan data realisasi impor beras beberapa periode sebelumnya seperti yang telah dijelaskan pada sub bab 2.2.6.1.

### **3.2.10. Membuat Fungsi Transfer Polinomial**

Setelah menentukan neuron masukan, maka dapat dibangun fungsi transfer polinomial kuadrat sebanyak jumlah kombinasi dari variabel input yang telah dibentuk pada tahapan sebelumnya. Masukan pada tahapan ini adalah variabel input pada jaringan GMDH sedangkan keluarannya adalah fungsi transfer polinomial kuadrat yang menghubungkan antara variabel input dan variabel output. Bentuk persamaan fungsi transfer yang dimaksud pada seperti pada Persamaan 2.2.6. Untuk menentukan jumlah neuron keluaran pada lapisan selanjutnya dapat menggunakan fungsi dan persamaan seperti yang telah dijelaskan pada sub bab 2.2.6.2.

### 3.2.11. Estimasi Koefisien Fungsi Transfer

Setelah membuat fungsi transfer polinomial maka dilakukan estimasi untuk setiap koefisien fungsi transfer tersebut. Masukan pada tahapan ini adalah fungsi transfer polinomial yang telah terbentuk dan data pelatihan realisasi impor beras, sedangkan keluaran pada tahapan ini adalah nilai estimasi dari setiap koefisien fungsi transfer. Untuk melakukan estimasi, maka dilakukan dengan metode *least square* dengan menggunakan data pelatihan, selengkapnya dapat dilihat pada sub bab 2.2.6.4. Hasil estimasi fungsi transfer ini merupakan model jaringan GMDH yang nantinya digunakan untuk meramalkan periode selanjutnya.

### 3.2.12. Menyeleksi Variabel Keluaran

Setelah melakukan estimasi koefisien transfer maka variabel masukan disubstitusikan ke dalam fungsi transfer sehingga menghasilkan neuron atau variabel keluaran pada *hidden layer* yang pertama. Selanjutnya dilakukan seleksi variabel keluaran yang terletak pada *hidden layer* dengan data pengujian. Neuron keluaran tersebut yang memiliki nilai RMSE terendah, maka akan dihilangkan dari jaringan dan sisanya akan dilanjutkan sebagai neuron masukan untuk lapisan selanjutnya. Masukan pada tahapan ini adalah data pengujian realisasi impor beras, fungsi transfer yang sudah diestimasi koefisiennya, dan hasil neuron pada setiap lapisan. Sedangkan keluaran dari tahapan ini adalah mendapatkan variabel yang terbaik pada setiap lapisan. Kriteria pemilihan variabel terbaik didasarkan pada beberapa indeks kinerja yaitu RMSE (*Root Mean Square Error*) seperti pada Persamaan 2.2.9 Hasil neuron keluaran akhir yang didapatkan yaitu berupa hasil peramalan impor beras pada periode saat ini ( $Y_t$ ) yang didapatkan dengan nilai RMSE yang minimum.

### 3.2.13. Mengecek Kriteria Penambahan Lapisan Berhenti

Langkah ini untuk menguji apakah himpunan persamaan model dapat diperbaiki lebih lanjut. Jika tidak ada perbaikan RMSE dari lapisan saat ini ke lapisan selanjutnya, maka penambahan lapisan jaringan akan diberhentikan dan mendapatkan model

akhir terbaik yang akan digunakan untuk peramalan periode selanjutnya. Untuk mengetahui detail kriteria penambahan lapisan berhenti, dapat dilihat pada sub bab 2.2.6.5. Masukan dari tahapan ini adalah nilai RMSE setiap neuron yang terpilih pada setiap lapisan, sedangkan keluarannya adalah mendapatkan keputusan apakah penambahan layer perlu dilanjutkan atau diberhentikan.

#### **3.2.14. Peramalan Data**

Setelah membuat model GMDH dengan nilai RMSE yang optimum, maka tahapan selanjutnya adalah meramalkan data intermiten untuk periode selanjutnya dengan menerapkan model GMDH yang telah dibentuk. Masukan pada tahapan ini adalah berupa model GMDH terbaik, sedangkan keluarannya adalah berupa hasil peramalan untuk periode selanjutnya. Model GMDH yang akan digunakan untuk peramalan yaitu berupa arsitektur jaringan GMDH dimana terdapat variabel masukan dan neuron yang relevan seperti yang telah dijelaskan pada Gambar 2.3 dan fungsi transfer yang sudah diestimasi untuk meminimumkan nilai RMSE.

#### **3.2.15. Analisa Hasil dan Penarikan Kesimpulan**

Setelah didapatkan hasil peramalan menggunakan metode GMDH, maka dapat dilakukan analisa hasil peramalan. Analisa hasil dapat dilakukan dengan memperhatikan nilai akhir dari RMSE dari pemodelan GMDH yang sudah dibentuk. Berdasarkan RMSE tersebut dapat ditarik kesimpulan apakah model GMDH yang dibentuk sudah dapat memberikan hasil peramalan yang lebih baik atau sebaliknya.

#### **3.2.16. Penulisan Laporan Tugas Akhir**

Setelah melakukan semua tahapan tersebut di atas, maka dilakukan penulisan laporan tugas akhir. Laporan ini digunakan sebagai dokumentasi terstruktur dari pengerjaan tugas akhir dan disusun berdasarkan konten dan format penulisan yang telah ditentukan.

### **3.3. Bahan dan Alat yang digunakan**

Pada sub bab ini dijelaskan mengenai bahan dan alat yang digunakan untuk pengerjaan tugas akhir. Bahan merupakan data-data atau objek terkait permasalahan yang diangkat, sedangkan alat merupakan perangkat lunak dan perangkat keras yang dibutuhkan untuk dapat mendukung pengerjaan tugas akhir.

#### **3.3.6. Bahan yang digunakan**

Data yang digunakan pada tugas akhir ini yaitu data impor beras wilayah Jawa Timur yang diperoleh dari Bulog Jawa Timur. Data tersebut diperoleh dalam bentuk format file .xlsx

#### **3.3.7. Alat yang digunakan**

Dalam mengerjakan tugas akhir ini alat utama yaitu sistem operasi Windows 10 dengan perangkat keras laptop. Perangkat lunak yang digunakan MATLAB R2013a sebagai alat untuk menyelesaikan permasalahan peramalan data intermiten dengan menggunakan metode GMDH dan Microsoft Excel 2016 untuk melakukan analisa model GMDH yang dibentuk.



## **BAB IV PERANCANGAN**

Bab ini menjelaskan mengenai perancangan model yang dibentuk menggunakan model atau struktur jaringan GMDH sebelum melakukan peramalan. Model jaringan yang akan dibentuk disesuaikan dengan langkah-langkah yang telah dijelaskan pada Bab Metodologi. Rancangan yang dibutuhkan yaitu meliputi proses pengumpulan data, penentuan variabel masukan dan keluaran, serta proses yang akan dilakukan dengan menggunakan perangkat lunak Matlab R2013a.

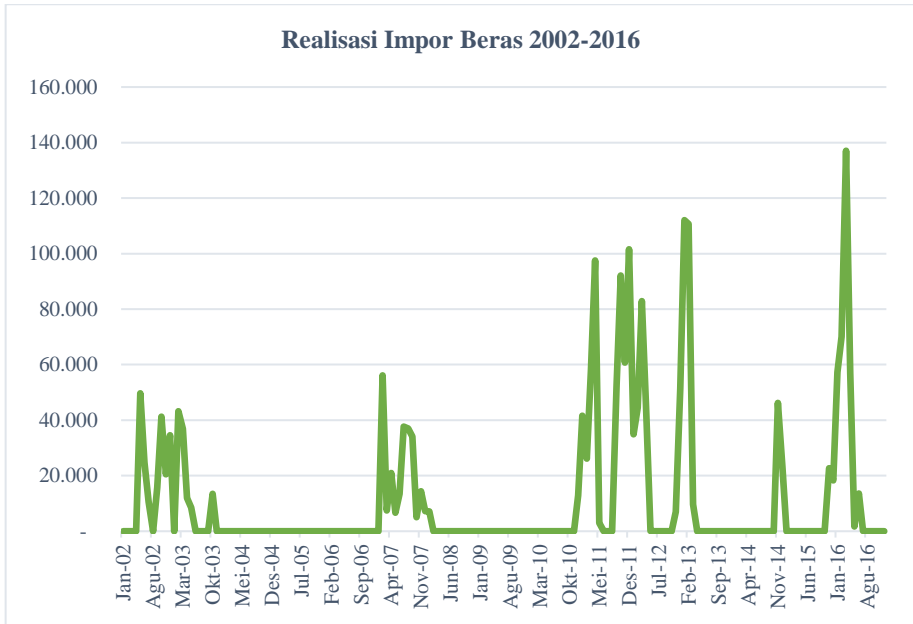
### **4.1. Pengumpulan dan *Pra-Processing* Data**

Sebelum membentuk model jaringan GMDH maka dilakukan pengumpulan dan *pra-processing* data.

#### **4.1.1. Pengumpulan Data**

Data yang digunakan pada tugas akhir ini adalah menggunakan data realisasi impor beras periode bulanan sejak tahun 2002 hingga 2016, sehingga diperoleh sebanyak 180 data poin observasi. Data ini didapatkan dari instansi Bulog wilayah Jawa Timur dengan kondisi banyak periode yang tidak dilakukan impor beras, sehingga dapat dikatakan data mentah yang diperoleh adalah data intermiten.

Data yang didapatkan dari Bulog tersebut disusun secara *time series* dari tahun 2002 hingga 2016. Kemudian direpresentasikan dengan grafik garis seperti pada Gambar 4.1



**Gambar 4.1 Data Realisasi Impor Beras 2002-2016**

Dapat dilihat pada Gambar 4.1 data realisasi impor beras terjadi secara terputus-putus karena terdapat beberapa atau banyak periode dimana Bulog Jawa Timur tidak melakukan impor beras, sehingga data yang tercatat adalah 0. Hal ini disebut sebagai data intermiten. Dengan demikian data realisasi impor beras ini sudah tepat untuk digunakan sebagai bahan pengerjaan tugas akhir.

#### **4.1.2. *Pra-Processing Data***

Setelah mendapatkan data, maka dilakukan *pra-processing* data. Tahapan ini melakukan transformasi data menggunakan log seperti pada Persamaan 2.2.1. Kemudian dilakukan pembagian data menjadi data *training* (pelatihan) dan data *testing* (pengujian) dengan persentase data *training* 85% dan data *testing* 15% dari total observasi data. Data pelatihan digunakan untuk melakukan estimasi terhadap koefisien fungsi

transfer yang dibentuk, sedangkan data pengujian digunakan untuk membuang atau menyeleksi variabel keluaran pada lapisan tersembunyi dengan menggunakan parameter MSE.

#### **4.2. Penentuan Variabel dan Parameter**

Dalam membuat model jaringan GMDH sebelum merumuskan fungsi transfer, hal yang perlu dilakukan adalah menentukan variabel-variabel yang terlibat dalam proses pembuatan model jaringan GMDH. Variabel yang diperlukan dalam jaringan yaitu variabel input, variabel output pada lapisan tersembunyi, dan variabel output akhir.

##### **4.2.1. Variabel Masukan (Input)**

Tugas akhir ini melakukan peramalan *time series* sehingga variabel input yang dibutuhkan yaitu sebanyak satu variabel, dan bukan peramalan kausalitas yang membutuhkan lebih dari satu variabel. Variabel input yang digunakan sebagai masukan pada jaringan GMDH adalah beberapa periode sebelumnya, dengan asumsi bahwa realisasi impor beras pada periode ini dipengaruhi oleh beberapa periode sebelumnya sebanyak  $t$  periode. Penamaan variabel input yang akan digunakan pada pengerjaan tugas akhir ini, yaitu:

$x_i$  = banyaknya beras yang diimpor pada periode  $y_{t-i}$  dalam satuan ton

Nilai  $i$  merupakan nilai maksimum lag yang akan diinputkan.

##### **4.2.2. Variabel Keluaran pada Lapisan Tersembunyi**

Variabel keluaran atau neuron yang dihasilkan untuk setiap lapisan tersembunyi, merupakan hasil dari fungsi transfer polinomial kuadrat. Banyaknya variabel keluaran pada lapisan tersembunyi ini bergantung pada banyaknya kombinasi neuron atau variabel input pada lapisan sebelumnya. Penamaan variabel keluaran pada lapisan tersembunyi, yaitu:

$z_{ij}$  = nilai neuron output ke- $i$  pada lapisan ke- $j$

#### 4.2.3. Variabel Keluran model jaringan GMDH

Variabel keluaran akhir pada jaringan GMDH yaitu hanya satu variabel yaitu periode saat ini. Penamaan variabel keluaran akhir pada jaringan GMDH, yaitu:

$F_t$  = hasil peramalan banyaknya realisasi pada periode ke- $t$  dalam satuan ton

#### 4.2.4. Menentukan inisiasi parameter

Dalam membangun jaringan GMDH selain variabel input, juga diperlukan inisiasi parameter yang dijadikan sebagai input dari jaringan. Inisiasi yang diperlukan yaitu inisiasi *lag input* dan jumlah maksimum neuron pada setiap lapisan. Nilai *lag input* atau jumlah neuron *input* diartikan sebagai pernyataan bahwa periode  $t$  dipengaruhi oleh beberapa periode sebelumnya sebanyak *lag input* yang ditentukan. Misalnya jika menentukan nilai lag input adalah 4, maka hasil peramalan pada periode  $Y_t$  dipengaruhi oleh  $(Y_{t-1})$ ,  $(Y_{t-2})$ ,  $(Y_{t-3})$ , dan  $(Y_{t-4})$ .

##### a) Maksimum Neuron pada Setiap *Hidden Layer*

Jumlah maksimum neuron pada setiap lapisan juga ditentukan untuk mendapatkan nilai neuron terbaik pada setiap lapisan. Nilai maksimum neuron bergantung pada jumlah kombinasi lag *input* atau neuron *input* seperti yang telah dijelaskan pada Persamaan 2.2.5. Pengaruh yang dihasilkan dari maksimum neuron adalah semakin besar nilai maksimum neuron yang ditentukan, maka akan semakin mendapatkan nilai RMSE yang besar. Sebaliknya jika nilai maksimum neuron yang ditentukan semakin kecil, maka akan dapat membuang neuron yang kurang relevan sehingga sehingga memungkinkan memperoleh nilai RMSE yang optimal [23].

##### b) Maksimum Lapisan pada Jaringan GMDH

Jumlah maksimum layer pada jaringan GMDH yang dibangun juga diinisiasi. Hal ini akan dilakukan uji coba dengan

menginputkan penambahan angka maksimum lapisan untuk mendapatkan *error* yang minimum pada keluaran akhir jaringan GMDH. Pada dasarnya, pengaruh yang dihasilkan dari maksimum layer adalah semakin besar nilai angka maksimum layer yang diinputkan, maka akan menghasilkan nilai RMSE pada *output* akhir yang semakin minimum karena pada setiap penambahan layer ada perbaikan RMSE. Sebaliknya jika nilai angka maksimum layer yang diinputkan semakin rendah, maka akan didapatkan nilai RMSE yang lebih besar karena kurangnya penambahan lapisan untuk perbaikan nilai RMSE pada keluaran akhir GMDH [23]. Dalam hal ini uji coba dilakukan berdasarkan penelitian sebelumnya yang telah dijelaskan pada dasar teori sub bab 2.2.6.5 paragraf 3 yaitu menggunakan inisiasi maksimum lapisan mulai 10 hingga 50 lapisan.

#### c) *Selection Pressure* ( $\alpha$ )

Parameter *selection pressure* ( $\alpha$ ) harus ditetapkan sebagai *threshold* yang tepat untuk menentukan jumlah neuron di setiap lapisan. Ambang batas ini akan mempengaruhi hasil akhir dari jaringan GMDH. Nilai seleksi yang digunakan yaitu ( $0 \leq \alpha \leq 1$ ). Nilai minimum dari seleksi yaitu 0 atau tidak ada tekanan dan nilai 1 untuk maksimum tekanan yang diberikan. Setelah menghitung koefisien untuk semua neuron, neuron yang menghasilkan hasil paling buruk dan melebihi kriteria seleksi yang dipilih sebelumnya, maka harus dikeluarkan dari lapisan. Dalam hal ini uji coba dilakukan berdasarkan penelitian sebelumnya yang telah dijelaskan pada dasar teori sub bab 2.2.6.5 paragraf 2 yaitu menggunakan nilai ambang batas  $0 \leq \alpha \leq 1$ .

### 4.3. Perumusan Fungsi Transfer

Fungsi transfer yang dibentuk pada jaringan GMDH merupakan penjemabatan antara variabel input dan variabel output pada setiap lapisan. Banyaknya neuron keluaran pada lapisan selanjutnya akan sama dengan banyaknya fungsi transfer yang

dirumuskan. Fungsi transfer yang digunakan yaitu polinomial kuadrat.

#### **4.4. Estimasi koefisien Fungsi Transfer**

Estimasi koefisien fungsi transfer polinomial kuadrat menggunakan teknik *least squared method*. Metode ini digunakan untuk meminimalkan RMSE dengan menggunakan teknik regresi bertingkat *non-linear*. Hal ini dilakukan dengan menggunakan fungsi regresi bertingkat *non-linear* pada matlab. Data yang digunakan untuk estimasi adalah data training dengan presentase 85% dari total data observasi keseluruhan.

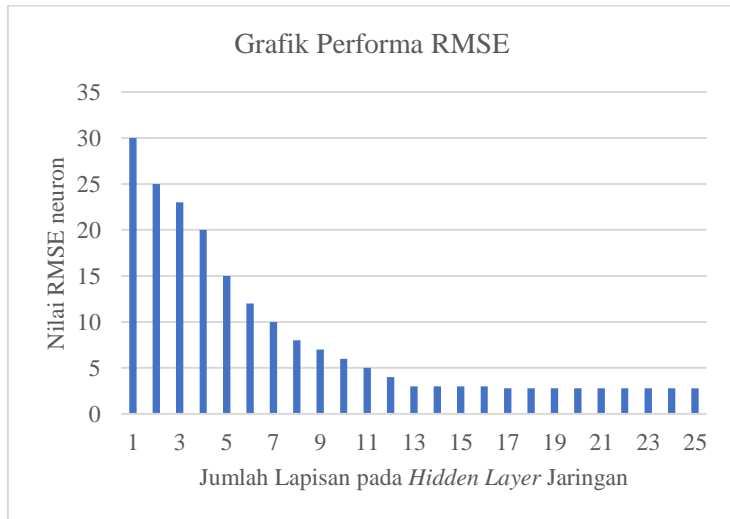
#### **4.5. Menyeleksi Variabel Keluran**

Setelah dilakukan estimasi terhadap semua fungsi transfer, maka variabel pada fungsi transfer disubstitusikan dengan nilai variabel *input* dua kombinasi yang dibentuk, sehingga didapatkan neuron pada lapisan berikutnya yang merupakan fungsi dari variabel input pada lapisan saat ini. Setiap neuron dihitung RMSE-nya dengan meninjau nilai output neuron dan output yang diharapkan (*target*). Neuron yang dihasilkan akan diurutkan berdasarkan nilai RMSE neuron yang terkecil hingga terbesar dengan menggunakan fungsi sortir yang terdapat pada MATLAB.

Neuron akan diseleksi dengan kriteria tertentu, yaitu nilai RMSE neuron yang diatas ambang batas (*threshold*) akan dibuang. Nilai ambang batas yang digunakan pada penelitian ini yaitu dengan nilai RMSE terbaik pada lapisan sebelumnya, sehingga apabila terdapat nilai RMSE neuron lebih besar dari pada nilai RMSE terbaik pada lapisan sebelumnya maka neuron tersebut dibuang. Apabila tidak terdapat neuron dengan RMSE yang melebihi ambang batas, maka program akan mengeksekusi variabel perintah maksimum neuron pada setiap lapisan yang dibentuk. Nilai RMSE yang digunakan untuk menyeleksi neuron adalah menggunakan RMSE neuron dari data pengujian dengan persentase 15% dari total data pelatihan.

#### 4.6. Mengecek Kriteria Penambahan Lapisan Berhenti

Setelah melakukan sortir RMSE neuron output dan membuang neuron yang memiliki RMSE lebih besar, maka RMSE minimum dari neuron yang terdapat pada lapisan saat ini dibandingkan dengan minimum nilai RMSE neuron pada lapisan sebelumnya. Jika tidak ada perbaikan dari RMSE maka penambahan lapisan diberhentikan. Sebaliknya jika ada perbaikan maka neuron output yang terpilih dijadikan sebagai input untuk lapisan selanjutnya. Pada Gambar 4.2 merupakan grafik garis yang menunjukkan performa penurunan nilai RMSE pada saat melakukan penambahan lapisan perbaikan RMSE.



**Gambar 4.2 Grafik Performa MSE Neuron Output**

Jumlah penambahan lapisan yang optimal dapat dilihat pada lapisan ke-13 karena pada lapisan ke-14 dan seterusnya tidak ada penurunan RMSE yang signifikan atau nilai RMSE konstan. Dengan demikian berdasarkan contoh grafik pada Gambar 4.2 maka penambahan lapisan dapat diberhentikan

pada lapisan ke-13 atau maksimum lapisan yang dapat dijadikan model GMDH adalah 13.

#### **4.7. Validasi Model GMDH**

Dalam melakukan validasi model untuk menguji tingkat akurasi, yaitu dilakukan dengan menghitung nilai RMSE, MDA, dan R. Model yang memiliki nilai RMSE lebih rendah, nilai MDA dan *correlation factor* (R) yang mendekati nilai 1 maka model tersebut yang terpilih untuk dilakukan peramalan periode selanjutnya.

#### **4.8. Peramalan Data**

Setelah melakukan uji coba pada keseluruhan data (data training dan testing) maka mendapatkan nilai MSE yang optimum dan didapatkan model jaringan GMDH yang akan digunakan untuk meramalkan periode selanjutnya. Model jaringan GMDH didapatkan dari tahapan menentukan variabel input hingga mengecek kriteria penambahan lapisan berhenti dimana model dari jaringan GMDH yaitu melingkupi hal berikut:

- Inisiasi *lag input*
- Jumlah lapisan dan jumlah neuron pada setiap lapisan yang optimal
- Koefisien fungsi transfer yang menghubungkan *input* dan *output* yang relevan.



## BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai proses implementasi metode jaringan GMDH ke dalam *tools* MATLAB. Hal ini dilakukan berdasarkan urutan proses pada Bab Metodologi dan rancangan kode disesuaikan dengan perancangan yang sudah dilakukan pada Bab Perancangan.

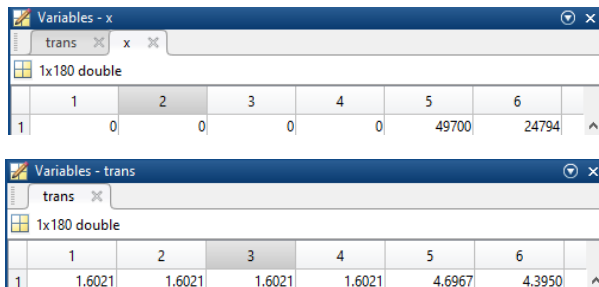
### 5.1. Pra-processing Data

Tahap *pra-processing* data merupakan tahap untuk melakukan transformasi data dan membagi data menjadi dua bagian, yaitu data pelatihan dan data pengujian yang akan diimplementasikan pada MATLAB. Sebelum melakukan pembagian data training dan testing, data ditransformasi menggunakan persamaan log seperti yang telah dijelaskan pada Persamaan 2.2.1. Segmen Kode Program 5.1 merupakan implementasi formula transformasi data menggunakan MATLAB

```
17 %% Load Data
18 data = load('impor');
19 x = data.x;
20 trans = log(x+40);
```

#### Segmen Kode Program 5.1 Implementasi Transformasi Data

Hasil perbandingan data aktual sebelum dan sesudah ditransformasi logaritma dapat dilihat pada Gambar 5.1:



The figure consists of two screenshots of the MATLAB 'Variables' window. The top screenshot, titled 'Variables - x', shows a 1x180 double array 'x'. The first six elements are displayed as 1, 0, 0, 0, 0, and 49700. The bottom screenshot, titled 'Variables - trans', shows the transformed data 'trans'. The first six elements are displayed as 1.6021, 1.6021, 1.6021, 1.6021, 4.6967, and 4.3950.

	1	2	3	4	5	6
x	1	0	0	0	0	49700

	1	2	3	4	5	6
trans	1.6021	1.6021	1.6021	1.6021	4.6967	4.3950

**Gambar 5.1 Contoh Hasil Transformasi Data Aktual**

Dapat dilihat pada Segmen Kode Program 5.2 baris ke-29 merupakan pendefinisian variabel jumlah persentase data pelatihan yang diinputkan. Variabel yang digunakan untuk mendefinisikan persentase data pelatihan yaitu pTrain dengan pembagian data pelatihan sebanyak 85% dari keseluruhan total data. Sedangkan variabel pTest merupakan variabel untuk mendefinisikan persentase data testing yang akan digunakan untuk validasi model jaringan GMDH. Variabel pTest merupakan dapat didefinisikan dengan  $1 - pTrain$ .

```

28      % Train Data
29 -    pTrain = 0.85;
30 -    nTrainData = round(pTrain*nData); %jumlah data training hasil pembulatan
31 -    TrainInd = Perm(1:nTrainData); %mendefinisikan train index
32 -    TrainInputs = Inputs(:,TrainInd);%mendefinisikan train input
33 -    TrainTargets = Targets(:,TrainInd);%mendefinisikan train target
34
35      % Test Data
36 -    pTest = 1 - pTrain;
37 -    nTestData = nData - nTrainData;
38 -    TestInd = Perm(nTrainData+1:end);
39 -    TestInputs = Inputs(:,TestInd);
40 -    TestTargets = Targets(:,TestInd);

```

### Segmen Kode Program 5.2 Pembagian Data Training dan Data Testing

Sedangkan variabel nTrainData dan nTestData merupakan jumlah roecord data pelatihan dan data pengujian dalam bilangan bulat, karena pada fungsi tersebut sudah terdapat fungsi round yang melakukan operasi pembulatan pada persentasi yang didefinisikan pada pTrain dikalikan dengan jumlah keseluruhan data. Variabel trainInd dan testInd merupakan variabel yang untuk mendapatkan index dari setiap data. Hal ini dilakukan untuk memudahkan dalam pembentukan neuron dan jaringan pada sistem. Variabel trainInput dan testInput digunakan untuk mendapatkan nilai input dari setiap record. Variabel trainTarget dan testTarget merupakan variabel target dari setiap neuron input yang akan dibentuk.

## 5.2. Menentukan Neuron Masukan

Dalam menentukan neuron masukan terdapat dua hal yang perlu dilakukan yaitu menentukan *lag input* dan inisiasi parameter

### 5.2.1. Menentukan Lag Input

Selanjutnya menentukan *lag input* dengan asumsi permintaan pada periode ke- $t$  akan dipengaruhi oleh beberapa periode sebelumnya. Pada Segmen Kode Program 5.3 telah dilakukan pendefinisian *lag input* dengan penamaan variabel *Delays* dalam bentuk matriks [1 2 3 4]. Arti dari pendefinisian ini yaitu hasil permintaan beras pada periode ke- $t$  akan dipengaruhi oleh 4 periode sebelumnya.

```

20 - x = data.x;
21 - n = size(x)
22
23 - Delays = [1 2 3 4]; %jumlah lag input
24 - [Inputs, Targets] = CreateTimeSeriesData(x, Delays);
25
26 - nData = size(Inputs, 2); %jumlah data keseluruhan.
27 - % Perm = randperm(nData);
28 - Perm = 1:nData; %index dari nData

```

#### Segmen Kode Program 5.3 Mendefinisikan Lag Input

Pendefinisian variabel *Inputs* dan *Targets* dideklarasikan dalam bentuk matriks [Inputs, Targets] dengan banyak lag input sebanyak 4. Jika periode ke-1 sampai ke-4 sebagai input maka nilai target yaitu terletak pada periode ke-5. Jika periode ke-2 sampai ke-5 sebagai input maka nilai target terletak pada periode ke-6, dan seterusnya sampai sebanyak data yang dimiliki. Jumlah data yang dipakai sebagai objek atau observasi tugas akhir ini dideklarasikan sebagai variabel *n* pada baris ke-21 pada Segmen Kode Program 5.3.

Variabel *input* dalam bentuk matriks [Inputs dan Targets] memanggil fungsi *CreateTimeSeriesData* yang memiliki parameter (*x*, *Delays*). Kode program untuk fungsi ini dibuat

file MATLAB baru dengan nama `CreateTimeSeriesData.m`. seperti pada Segmen Kode Program 5.4

```

13 function [X, Y] = CreateTimeSeriesData(x, Delays)
14 -     T = size(x,2);
15 -     MaxDelay = max(Delays);
16 -     Range = MaxDelay+1:T;
17 -     X= [];
18 -     for d = Delays
19 -         X=[X; x(:,Range-d)];
20 -     end
21 -     Y = x(:,Range);
22 - end

```

#### Segmen Kode Program 5.4 Membuat Matriks Neuron Input

Variabel `T` merupakan jumlah  $n$  data keseluruhan saat sebelum diinisiasi *lag input*, yaitu sebanyak 180 data. Variabel `MaxDelay` merupakan angka maksimum dari matriks [1 2 3 4] atau menyatakan jumlah *lag input* periode. Jadi nilai dari `MaxDelay` tersebut adalah 4.

#### 5.2.2. Inisiasi Parameter

Inisiasi parameter dilakukan untuk mendefinisikan jumlah maksimum neuron pada jaringan GMDH, jumlah maksimum layer, dan nilai parameter.

```

%% Create and Train GMDH Network

params.MaxLayerNeurons = 15; % Maximum Number of Neurons in a Layer
params.MaxLayers = 10;      % Maximum Number of Layers
params.pTrain=0.85;         % Train Ratio
params.alpha=0.1;           %selection pressure
gmdh = GMDH(params, TrainInputs, TrainTargets);

```

#### Segmen Kode Program 5.5 Inisiasi Parameter Jaringan GMDH

Pada Segmen Kode Program 5.5 ditunjukkan terdapat beberapa variabel untuk menginisiasi maksimum neuron yaitu pada variabel `params.MaxLayerNeuron`, variabel untuk menginisiasi maksimum layer pada jaringan yaitu `params.MaxLayers`, variabel `params.alpha` yaitu untuk mendefinisikan tingkat

tekanan seleksi yang akan diimplementasikan pada proses training data. Variabel `params.pTrain` yaitu digunakan untuk membagi data pelatihan dan pengujian. Data pelatihan pada variabel ini dengan proporsi 85% digunakan untuk melakukan estimasi terhadap bobot atau koefisien pada fungsi transfer, sedangkan sebanyak 15% digunakan untuk menyeleksi neuron yang akan digunakan sebagai input pada lapisan selanjutnya.

Setelah menentukan atau inisiasi keempat parameter tersebut maka dilakukan proses pelatihan jaringan GMDH. Hal ini diimplementasikan pada MATLAB pada file `main.m` seperti pada Segmen Kode Program 5.5 dengan penamaan variabel `gmdh` membentuk fungsi GMDH dengan parameter (`params`, `TrainInputs`, `TrainTargets`). Fungsi `gmdh` akan memanggil parameter-parameter yang sudah diinisiasikan pada Segmen Kode Program 5.6

```

14  function gmdh = GMDH(params, X, Y)
15  -      MaxLayerNeurons = params.MaxLayerNeurons;
16  -      MaxLayers = params.MaxLayers;
17  -      alpha = params.alpha;
18  -      nData = size(X,2);

```

#### Segmen Kode Program 5.6 Fungsi Pelatihan Jaringan GMDH

Fungsi `gmdh` ini merupakan fungsi untuk memulai proses pelatihan menggunakan data pelatihan yang sudah ditentukan pada proses sebelumnya. Kemudian mendefinisikan variabel `pTrainData` pada Segmen Kode Program 5.7 baris ke-26 dengan memanggil parameter yang sudah ditentukan sebelumnya yaitu `pTrain` senilai 85%.

```

25      % Divide Data
26      pTrainData = params.pTrain;
27      nTrainData = round(pTrainData*nData);
28      X1 = X(:,1:nTrainData); %data training input jaringan GMDH
29      Y1 = Y(:,1:nTrainData); %data training output jaringan GMDH
30      pTestData = 1-pTrainData;
31      nTestData = nData - nTrainData;
32      X2 = X(:,nTrainData+1:end); %data testing input jaringan GMDH
33      Y2 = Y(:,nTrainData+1:end); %data training output jaringan GMDH
34      Layers = cell(MaxLayers, 1);
35      Z1 = X1;
36      Z2 = X2;

```

```

% Divide Data
pTrainData = params.pTrain;
nTrainData = round(pTrainData*nData);
X1 = X(:,1:nTrainData); %data training
input jaringan GMDH
Y1 = Y(:,1:nTrainData); %data training
output jaringan GMDH
pTestData = 1-pTrainData;
nTestData = nData - nTrainData;
X2 = X(:,nTrainData+1:end); %data testing
input jaringan GMDH
Y2 = Y(:,nTrainData+1:end); %data training
output jaringan GMDH
Layers = cell(MaxLayers, 1);
Z1 = X1;
Z2 = X2;

```

#### Segmen Kode Program 5.7 Pembagian data training dan testing pada proses pelatihan jaringan

Kemudian dilakukan pembagian data menjadi X1 dan Y1 yang merupakan nilai dari *input* dan target dari data pelatihan serta X2 dan Y2 yang merupakan nilai dari *input* dan target dari data pengujian. Variabel X1 dan Y1 akan digunakan untuk melakukan estimasi terhadap koefisien fungsi transfer polinomial. Sedangkan variabel X2 dan Y2 akan digunakan untuk melakukan seleksi neuron terbaik yang nantinya akan menjadi *input* dari lapisan selanjutnya. Variabel Z1 merupakan neuron *output* atau hasil *output* dari X1 pada lapisan sebelumnya yang akan dijadikan sebagai inputan baru untuk lapisan selanjutnya, sehingga variabel Z1 dijadikan sebagai variabel X1 atau  $Z1=X1$ . Begitu juga untuk Z2 yang merupakan

hasil *output* dari data pengujian ( $X_2$ ) pada lapisan sebelumnya yang akan dijadikan sebagai inputan baru untuk lapisan selanjutnya, sehingga variabel  $Z_2$  dijadikan sebagai variabel  $X_2$  atau  $Z_2 = X_2$

### 5.3. Perumusan Fungsi Transfer

Dalam melakukan pelatihan terhadap data pelatihan jaringan GMDH, maka dibentuklah fungsi transfer yang menjembatani antara lapisan *input* dan lapisan neuron *output*. Hal ini dilakukan untuk setiap lapisan hingga mencapai lapisan maksimum yang telah ditentukan, seperti yang telah diimplementasikan pada kode MATLAB pada Segmen Kode Program 5.8

39 -	<code>L = GetPolynomialLayer(Z1, Y1, Z2, Y2);</code>
------	------------------------------------------------------

#### Segmen Kode Program 5.8 Variabel Pembentukan Lapisan Jaringan GMDH

Variabel  $L$  merupakan variabel *layer* atau lapisan yang akan dibentuk dengan memanggil fungsi `GetPolynomialLayer`. Parameter yang digunakan pada fungsi ini yaitu data *training input* ( $Z_1$ ), data *training target* ( $Y_1$ ), data *testing input* ( $Z_2$ ), dan data *testing target* ( $Y_2$ ).

Kemudian dibangunlah fungsi `GetPolynomialLayer` dengan membuat file fungsi baru yaitu `GetPolynomialLayer.m` yang dapat dilihat pada Segmen Kode Program 5.9. Isi dari fungsi tersebut adalah untuk mendapatkan hasil neuron *output* pada lapisan selanjutnya. Jumlah neuron *output* yang dihasilkan pada lapisan selanjutnya merupakan sebanyak jumlah kombinasi dari neuron *input*, hal ini diimplementasikan dengan variabel  $N$  pada baris ke-16. Dalam membangun neuron ini tentunya terdapat fungsi transfer polinomial kuadrat non-linier yang koefisiennya diestimasi menggunakan data training. Untuk melakukan estimasi tersebut, maka dapat dilakukan *fitting* polinomial pada baris ke-20 hingga ke-25 pada Segmen Kode Program 5.9:

```

14 function L = GetPolynomialLayer(X1, Y1, X2, Y2)
15     n = size(X1,1); %jumlah input untuk next layer
16     N = n*(n-1)/2; %menghitung jumlah neuron selanjutnya
17     template = FitPolynomial(rand(2,3),rand(1,3),rand(2,3),rand(1,3),[]);
18     L = repmat(template, N, 1);
19     k = 0;
20     for i=1:n-1
21         for j=i+1:n
22             k = k+1;
23             L(k) = FitPolynomial(X1([i j],:), Y1, X2([i j],:), Y2, [i j]);
24         end
25     end
26     [~, SortOrder] = sort([L.RMSE2]);
27     L = L(SortOrder);
28 end

```

```

function L = GetPolynomialLayer(X1, Y1, X2,
Y2)
    n = size(X1,1); %jumlah input untuk next
layer
    N = n*(n-1)/2; %menghitung jumlah neuron
selanjutnya
    template =
FitPolynomial(rand(2,3),rand(1,3),rand(2,3),ra
nd(1,3),[]);
    L = repmat(template, N, 1);
    k = 0;
    for i=1:n-1
        for j=i+1:n
            k = k+1;
            L(k) = FitPolynomial(X1([i j],:),
Y1, X2([i j],:), Y2, [i j]);
        end
    end
    [~, SortOrder] = sort([L.RMSE2]);
    L = L(SortOrder);
end

```

### Segmen Kode Program 5.9 Fungsi Pembentukan Lapisan Jaringan GMDH

Pada perulangan tersebut dimaksudkan untuk dapat melakukan estimasi fungsi transfer polinomial pada semua neuron input pada setiap lapisan. Variabel  $L(k)$  merupakan variabel yang dibentuk untuk memanggil fungsi `FitPolynomial` dengan parameter data *training input* (X1) untuk semua neuron pada



setiap layer, data *training* target (Y1), data *testing input* (X2) untuk semua neuron pada setiap layer, data *testing* target (Y2), dan untuk semua indeks perulangan i dan j atau (untuk semua neuron pada setiap lapisan).

#### 5.4. Estimasi Koefisien Fungsi Transfer

Kemudian dibangunlah fungsi FitPolynomial dengan membuat file fungsi baru yaitu FitPolynomial.m yang dapat dilihat pada Segmen Kode Program 5.10:

```

14 function p = FitPolynomial(x1, Y1, x2, Y2, vars)
15     % x1 =data training, x2=data testing
16     X1 = CreateRegressorsMatrix(x1); %membuat matrix regresi
17     c = Y1*pinv(X1); %menentukan koefisien fungsi transfer
18     Ylhat = c*X1; %hasil output neuron data training
19     e1 = Y1- Ylhat; %error data training
20     MSE1 = mean(e1.^2);
21     RMSE1 = sqrt(MSE1);

```

#### Segmen Kode Program 5.10 Mengestimasi Koefisien Fungsi Transfer

Dalam membuat fungsi untuk estimasi koefisien yaitu menggunakan teknik *multiple non-linier regression*. Dalam mengimplementasikan teknik tersebut, maka perlu menyusun matrik regresi terlebih dahulu seperti pada baris ke-16 Segmen Kode Program 5.10. Variabel c pada baris ke-17 merupakan koefisien yang didapatkan untuk setiap variabel dari fungsi transfer yang nantinya dijadikan sebagai model untuk dilakukan validasi dan peramalan periode selanjutnya.

Setelah dilakukan estimasi koefisien fungsi transfer, maka neuron *input* disubstitusikan pada fungsi transfer polinomial yang diimplementasikan pada variabel Ylhat tersebut, sehingga Ylhat merupakan hasil neuron *output* data *training* dari fungsi transfer polinomial yang sudah dilakukan estimasi koefisien. Hasil neuron *output training* ini (Ylhat) akan dibandingkan dengan nilai *output training* target (Y1) dan dihitung residu atau nilai *error* yang telah dilakukan pada baris ke-19 pada Segmen

Kode Program 5.10. Dari nilai *error* tersebut dapat dihitung nilai MSE (*Mean Square Error*) untuk data training yang telah diimplementasikan dengan menggunakan variabel MSE1, sedangkan RMSE1 merupakan implementasi untuk menghitung nilai *Root Mean Square Error* dari data training.

Setelah mengimplementasikan atau men-subtitusikan fungsi transfer pada *training*, juga disubtitusikan pada data *testing*. Hal ini sudah dilakukan dengan menggunakan variabel Y2hat pada baris ke-24 Segmen Kode Program 5.11

```

23 -      f = @(x) c*CreateRegressorsMatrix(x);
24 -      Y2hat = f(x2); %hasil output neuron data testing
25 -      e2 = Y2- Y2hat; %error data testing
26 -      MSE2 = mean(e2.^2);
27 -      RMSE2 = sqrt(MSE2);
--

```

#### **Segmen Kode Program 5.11 Subtitusi Estimasi Fungsi Transfer terhadap Data Testing**

Setelah itu akan dibandingkan neuron *output* data *testing* tersebut (Y2hat) dengan target data *testing* (Y2) dan dihitung residu atau nilai *error* yang telah dilakukan pada baris ke-26 Segmen Kode Program 5.11. Dari nilai *error* tersebut dapat dihitung nilai MSE (*Mean Square Error*) untuk data *testing* yang telah diimplementasikan dengan menggunakan variabel MSE2, sedangkan RMSE2 merupakan implementasi untuk menghitung nilai *Root Mean Square Error* dari data *testing*.

Fungsi transfer yang digunakan untuk menghasilkan neuron output menggunakan polinomial kuadrat 2 variabel, sehingga fungsi dapat dituliskan seperti pada Segmen Kode Program 5.12:

```

39 - function X = CreateRegressorsMatrix(x)
40 -     X = [ones(1,size(x,2))
41           x(1,:)
42           x(2,:)
43           x(1,:).^2
44           x(2,:).^2
45           x(1,:).*x(2,:)];
46 - end

```

### Segmen Kode Program 5.12 Perumusan Fungsi Transfer Menggunakan Polinomial Kuadrat Dua Variabel

Pada Segmen Kode Program 5.12 tersebut merupakan pembentukan regresi polinomial kuadrat dengan 2 variabel *input*. *x1* dan *x2* merupakan kombinasi variabel *input* sesuai dengan *lag input* periode yang ditentukan.

### 5.5. Menyeleksi Variabel Keluaran

Neuron *output* yang dihasilkan dari fungsi transfer akan dihitung RMSE dari data pelatihan dan data pengujian. Kemudian nilai RMSE tersebut disortir menggunakan data testing, karena fungsi data testing pada jaringan ini adalah untuk menyeleksi neuron yang akan dijadikan sebagai input untuk lapisan selanjutnya. Hal ini telah diimplementasikan pada baris ke-26 dan ke-27 pada Segmen Kode Program 5.13

```

14 - function L = GetPolynomialLayer(X1, Y1, X2, Y2)
15 -     n = size(X1,1); %jumlah input untuk next layer
16 -     N = n*(n-1)/2; %menghitung jumlah neuron selanjutnya
17 -     template = FitPolynomial(rand(2,3),rand(1,3),rand(2,3),rand(1,3),[]);
18 -     L = repmat(template, N, 1);
19 -     k = 0;
20 -     for i=1:n-1
21 -         for j=i+1:n
22 -             k = k+1;
23 -             L(k) = FitPolynomial(X1([i j],:), Y1, X2([i j],:), Y2, [i j]);
24 -         end
25 -     end
26 -     [~, SortOrder] = sort([L.RMSE2]);
27 -     L = L(SortOrder);
28 - end

```

### Segmen Kode Program 5.13 Pembentukan Lapisan Jaringan GMDH dan Pengurutan Neuron

Nilai RMSE neuron *output* yang melebihi batas maksimum neuron, maka akan dibuang, sedangkan neuron yang lolos dengan nilai RMSE yang minimum akan dijadikan neuron input untuk penambahan lapisan selanjutnya. Hal ini dilakukan secara berulang hingga tidak ada penurunan RMSE antar lapisan.

### 5.6. Mengecek Kriteria Penambahan Lapisan Berhenti

Perulangan jaringan GMDH terletak pada penambahan lapisan untuk meminimumkan *error* dari neuron output. Hal ini dipengaruhi oleh parameter maksimum *layer* yang telah ditentukan. Dalam implementasi pengecekan penambahan lapisan dilihat pada hasil minimum *error* pada setiap lapisan. Apabila dari lapisan saat ini ke lapisan selanjutnya tidak ada perbaikan *error*, maka penambahan lapisan berhenti pada lapisan tersebut. Segmen Kode Program 5.14 menunjukkan perulangan untuk penambahan layer yang disesuaikan dengan maksimum lapisan yang telah ditentukan. Jika proses pelatihan melebihi batas, maka output akan tetap dibatasi sesuai dengan maksimum lapisan tersebut.

```

37 - for l = 1:MaxLayers
38 -     L = GetPolynomialLayer(Z1, Y1, Z2, Y2);
39 -     ec = alpha*L(1).RMSE2 + (1-alpha)*L(end).RMSE2;
40 -     ec = max(ec, L(1).RMSE2);
41 -     L = L([L.RMSE2] <= ec);
42 -     if numel(L) > MaxLayerNeurons %membatasi jumlah neuron sesuai threshold
43 -         L = L(1:MaxLayerNeurons);
44 -     end
45 -     if l==MaxLayers && numel(L)>1
46 -         L = L(1);
47 -     end
48 -     Layers{l} = L;
49 -     Z1 = reshape([L.Yihat],nTrainData,[]);
50 -     Z2 = reshape([L.Yihat],nTestData,[]);
51 -     disp(['Layer ' num2str(l) ': Neurons = ' num2str(numel(L)) ', Min Error = ' num2str(L(1).RMSE2)]);
52 -     %disp([num2str(L(1).RMSE2)]);
53 -     if numel(L)==1
54 -         break;
55 -     end
56 - end

```

```

for l = 1:MaxLayers

    L = GetPolynomialLayer(Z1, Y1, Z2,
        Y2);
    ec = alpha*L(1).RMSE2 + (1-
        alpha)*L(end).RMSE2;
    ec = max(ec, L(1).RMSE2);
    L = L([L.RMSE2] <= ec);

```

```

if numel(L) > MaxLayerNeurons
    L = L(1:MaxLayerNeurons);
end
if l==MaxLayers && numel(L)>1
    L = L(1);
end
Layers{l} = L;
Z1 =
reshape([L.Y1hat],nTrainData, [])';
Z2 =
reshape([L.Y2hat],nTestData, [])';
disp(['Layer ' num2str(l) ': Neurons = '
' num2str(numel(L)) ', Min Error = '
num2str(L(l).RMSE2)]);
if numel(L)==1
    break;
end
end

```

**Segmen Kode Program 5.14 Perulangan Penambahan Lapisan  
pada Pelatihan Jaringan GMDH**

### 5.7. Validasi Model GMDH

Setelah dilakukan proses pelatihan data, maka selanjutnya adalah proses validasi model. Proses validasi model dilakukan dengan membandingkan hasil *output* dan data terget pada data *training* dan data *testing*. Validasi menggunakan dua parameter yaitu RMSE (*Root Mean Square Error*) dan MDA (*Mean Directional Accuracy*) Untuk dapat mengetahui output akhir dari setiap periode peramalan menggunakan jaringan GMDH, maka dapat dilihat pada Segmen Kode Program 5.15

```

42      %% Evaluate GMDH Network and predict output
43 -    Outputs = ApplyGMDH(gmdh, Inputs);
44 -    TrainOutputs = Outputs(:,TrainInd);
45 -    TestOutputs = Outputs(:,TestInd);
46
47 -    InverseOutputs = (10.^ Outputs)-40;
48 -    InverseTrainOutputs = (10.^ TrainOutputs)-40;
49 -    InverseTestOutputs = (10.^ TestOutputs)-40;
50

```

### Segmen Kode Program 5.15 Pembentukan Neuron Output Akhir pada Setiap Periode

Pada Segmen Kode Program 5.15 tersebut merupakan perintah untuk menghasilkan *output* neuron akhir dari setiap periode bulanan. Pada baris ke-43 merupakan perintah untuk *output* neuron akhir semua periode. Kode ini memanggil fungsi `ApplyGMDH` yang berisi model yang telah dibentuk pada fungsi `GMDH` sebelumnya. Pada baris ke-44 dan ke-45 digunakan untuk mengetahui *output* neuron akhir periode bulanan untuk masing-masing data pelatihan dan pengujian. Baris ke 47-49 merupakan tahap detransformasi untuk mengembalikan hasil transformasi data ke bentuk semula.

Fungsi `ApplyGMDH` digunakan untuk mengimplementasikan parameter dan model jaringan `GMDH` yang telah dibentuk pada fungsi `GMDH.m`. Pada Segmen Kode Program 5.16 merupakan fungsi `ApplyGMDH` yang di dalam kode tersebut telah memanggil parameter jumlah maksimum layer dan maksimum neuron yang telah didefinisikan pada kode utama. Kemudian hasil *output* akhir yang dihasilkan diperoleh dari perulangan sebanyak maksimum lapisan dan maksimum neuron yang telah ditentukan. Hal ini telah diimplementasikan pada Segmen Kode Program 5.16

```

14 function Yhat = ApplyGMDH(gmdh, X)
15     nLayer = numel(gmdh.Layers); %jumlah layer
16     Z = X; %input times series
17     for l=1:nLayer
18         Z = GetLayerOutput(gmdh.Layers{l}, Z); %hasil output per layer per neuron
19     end
20     Yhat = Z;
21 end
22
23 function Z = GetLayerOutput(L, X)
24     m = size(X,2); %jumlah keseluruhan data
25     N = numel(L); %jumlah neuron
26     Z = zeros(N,m);
27     for k=1:N
28         vars = L(k).vars; %kombinasi variabel (index) untuk fungsi transfer
29         x = X(vars,:); % hasil output substitusi fungsi transfer
30         Z(k,:) = L(k).f(x);
31     end

```

```

function Yhat = ApplyGMDH(gmdh, X)
    nLayer = numel(gmdh.Layers); %jumlah
    layer
    Z = X; %input times series
    for l=1:nLayer
        Z = GetLayerOutput(gmdh.Layers{l},Z);
%hasil output per layer per neuron
    end
    Yhat = Z;
end
function Z = GetLayerOutput(L, X)
    m = size(X,2); %jumlah keseluruhan data
    N = numel(L); %jumlah neuron
    Z = zeros(N,m);
    for k=1:N
        vars = L(k).vars; %kombinasi variabel
(index) untuk fungsi transfer
        x = X(vars,:); % hasil output
subtitusi fungsi transfer
        Z(k,:) = L(k).f(x);
    end
end
end

```

### Segmen Kode Program 5.16 Implementasi Model Jaringan GMDH

Dengan adanya proses implementasi model jaringan terhadap data training dan data testing, maka model dapat divalidasi dengan membandingkan data output dan data target.

### 5.7.1. Menghitung RMSE

Dalam melakukan perbandingan data aktual dan data peramalan (keluaran dari jaringan GMDH) maka dilakukan validasi terhadap data training dan data testing menggunakan RMSE. Perhitungan RMSE yang dilakukan menggunakan MATLAB ditunjukkan pada Segmen Kode Program 5.17

```

14 function PlotResults(Targets, Outputs, Title)
15 -     Errors = Targets - Outputs;
16 -     MSE = mean(Errors.^2);
17 -     RMSE = sqrt(MSE)
18 -     ErrorMean = mean(Errors);

```

#### Segmen Kode Program 5.17 Perhitungan RMSE

Dengan dilakukan perhitungan tersebut, maka nantinya dapat dilakukan analisa perbandingan beberapa model untuk pemilihan model terbaik yang akan digunakan untuk meramalkan periode selanjunya

### 5.7.2. Menghitung MDA

Selain menggunakan parameter RMSE, juga dihitung akurasi MDA untuk mengetahui pola data peramalan apakah sesuai atau mengikuti pola data aktual. Contoh perhitungan MDA yang implementasikan menggunakan excel dapat dilihat pada Gambar 5.2:

	A	B	C	D	E
1	Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
5	0				
6	49700				
7	24794				
8	10600	30517,37662	0		
9	0	657,0383018	0	0	1
10	15800	535,7020893	1	0	0
11	41275	4080,352844	1	1	1
12	20450	16646,36569	0	1	0
13	34600	31337,33188	1	1	1
14	0	16248,5385	0	0	1
15	43243	1289,47651	1	0	0

Gambar 5.2 Contoh Perhitungan MDA



Kolom C diisi angka 1 jika hasil data aktual pada periode ke Yt lebih besar dari data pada periode ke Yt-1, dan diisi angka 0 untuk kondisi sebaliknya. Kolom D diisi angka 1 jika hasil data peramalan pada periode ke Ft lebih besar dari data pada periode ke Ft-1, dan diisi angka 0 untuk kondisi sebaliknya. Sedangkan atribut akurasi pada kolom E merupakan biimplikasi dari kolom C dan D. Jika C dan D terdiri dari angka yang sama maka kolom E diisi dengan angka 1, dan jika berbeda maka diisi angka 0. Nilai MDA yang diperoleh merupakan nilai rata-rata dari akurasi (kolom E) untuk semua periode.

### 5.7.3. Menghitung Correlation Factor (R)

*Correlation factor* dihitung untuk mengetahui pengaruh variabel target dengan output yang dihasilkan dari proses peramalan menggunakan metode jaringan GMDH. Segmen Kode Program 5.18 dilakukan *plotting* data untuk mengetahui hubungan atau pengaruh kedua variabel tersebut

```

89 - if ~isempty(which('plotregression'))
90 -     figure;
91 -     plotregression(TrainTargets, TrainOutputs, 'Train Data', ...
92 -                   TestTargets, TestOutputs, 'TestData', ...
93 -                   Targets, Outputs, 'All Data');
94 - end

```

**Segmen Kode Program 5.18 Implementasi Perhitungan  
Correlation Factor**

### 5.8. Plot Hasil Pemodelan

Setelah melakukan pemodelan dengan menggunakan data pelatihan dan melakukan proses validasi menggunakan data pengujian, maka selanjutnya melakukan plot data target dan output untuk masing-masing data training testing. Hal ini dilakukan dengan membuat grafik garis untuk mengetahui pola data dan dapat membandingkan secara langsung antara data target dan data output yang dihasilkan.

Pada Segmen Kode Program 5.19 merupakan kode program untuk melakukan plotting terhadap data pelatihan. Legenda dari grafik ini yaitu data pelatihan *output* dan data pelatihan target.

```
58 - figure;
59 - PlotResults(TrainTargets, TrainOutputs, 'Train Data');
```

#### **Segmen Kode Program 5.19 Plotting Target dan Output Data Training**

Pada Segmen Kode Program 5.20 merupakan kode program untuk melakukan plotting terhadap data pengujian. Legenda dari grafik ini yaitu data pengujian output dan data pengujian target.

```
61 - figure;
62 - PlotResults(TestTargets, TestOutputs, 'Test Data');
```

#### **Segmen Kode Program 5.20 Plotting Target dan Output Data Testing**

Pada Segmen Kode Program 5.21 merupakan kode program untuk melakukan plotting terhadap data pengujian dan pelatihan. Legenda dari grafik ini yaitu data output dan data target

```
64 - figure;
65 - PlotResults(Targets, Outputs, 'All Data');
```

#### **Segmen Kode Program 5.21 Plotting Target dan Output Data Training dan Testing**

Kode program untuk melakukan plotting seperti pada Segmen Kode Program 5.19, Segmen Kode Program 5.20, dan Segmen Kode Program 5.21 memanggil fungsi PlotResult. Fungsi ini memiliki parameter target, output, dan judul seperti yang telah diimplementasikan pada

Segmen Kode Program 5.22

```

14 - function PlotResults(Targets, Outputs, Title)
15 -     Errors = Targets - Outputs;
16 -     MSE = mean(Errors.^2);
17 -     RMSE = sqrt(MSE);
18 -     ErrorMean = mean(Errors);
19 -     ErrorStd = std(Errors);
20 -
21 -     subplot(2,2,[1 2]);
22 -     plot(Targets);
23 -     hold on;
24 -     plot(Outputs);
25 -     legend('Targets','Outputs');
26 -     ylabel('Targets and Outputs');
27 -     grid on;
28 -     title(Title);
29 -
30 -     subplot(2,2,3);
31 -     plot(Errors);
32 -     title(['MSE = ' num2str(MSE) ', RMSE = ' num2str(RMSE)]);
33 -     ylabel('Errors');
34 -     grid on;
35 -
36 -     subplot(2,2,4);
37 -     histfit(Errors, 50);
38 -     title(['Error Mean = ' num2str(ErrorMean) ', Error Std = ' num2str(ErrorStd)]);
39 - end

```

```

function PlotResults(Targets, Outputs, Title)
    Errors = Targets - Outputs;
    MSE = mean(Errors.^2);
    RMSE = sqrt(MSE);
    ErrorMean = mean(Errors);
    ErrorStd = std(Errors);
    subplot(2,2,[1 2]);
    plot(Targets);
    hold on;
    plot(Outputs);

    legend('Targets','Outputs');
    ylabel('Targets and Outputs');
    grid on;
    title(Title);
    subplot(2,2,3);
    plot(Errors);
    title(['MSE = ' num2str(MSE) ', RMSE = '
num2str(RMSE)]);
    ylabel('Errors');
    grid on;
    subplot(2,2,4);
    histfit(Errors, 50);
    title(['Error Mean = ' num2str(ErrorMean)
', Error Std = ' num2str(ErrorStd)]);
end

```

### Segmen Kode Program 5.22 Fungsi Plotting Data Training dan Testing pada Grafik Garis

### 5.9. Peramalan Data

Peramalan data dilakukan setelah tahap pemodelan dilakukan. Model yang didapatkan berupa variabel-variabel neuron yang relevan terhadap *output* dan koefisien fungsi transfer. Segmen Kode Program 5.23 merupakan kode untuk memuat data peramalan dan memanggil fungsi `CreateTimeSeriesPeramalan` untuk menyusun data berdasarkan *lag input* yang ditentukan.

```

58      %% Forecasting next period
59 -    dataPeramalan = load('peramalan');
60 -    x = dataPeramalan.x;
61
62 -    [InputsPeramalan] = CreateTimeSeriesPeramalan(x, Delays);
63
64 -    OutputsAllForecast = ApplyGMDH(gmdh, InputsPeramalan);
65 -    ForecastOutputs = OutputsAllForecast(:,165);

```

#### Segmen Kode Program 5.23 Peramalan Data untuk Beberapa Periode ke Depan

Parameter yang digunakan pada Segmen Kode Program 5.24 untuk menyusun matriks *input* sesuai dengan *lag input* yang ditentukan adalah *input* pada data peramalan yaitu variabel *x* dan variabel *Delay* sebagai *lag input* atau banyaknya periode sebelumnya yang mempengaruhi hasil peramalan pada periode sekarang.

```

13  function [X] = CreateTimeSeriesPeramalan(x, Delays)
14 -     T = size(x,2);
15 -     MaxDelay = max(Delays);
16 -     Range = MaxDelay:T;
17 -     X= [];
18 -     for d = Delays
19 -         X=[X; x(:,Range-d+1)];
20 -     end
21 - end

```

#### Segmen Kode Program 5.24 Membuat Matriks Neuron Input Data Peramalan

Pada Segmen Kode Program 5.23 juga memanggil fungsi `ApplyGMDH` untuk mendapatkan model yang sudah dibentuk

pada tahap pelatihan dan sudah divalidasi menggunakan data testing.

### 5.10. Plot Hasil Peramalan

Setelah melakukan peramalan sebanyak 24 periode bulanan atau 2 tahun, maka hasil dari peramalan tersebut di-plot untuk mengetahui pola data dari hasil peramalan periode selanjutnya. Hal ini telah diimplementasikan pada Segmen Kode Program 5.25

```
78 - figure;
79 - PlotResultsPeramalan(Targets, OutputsAllForecast, 'Forecast');
```

#### Segmen Kode Program 5.25 Plotting Target dan Output Data Peramalan

Pada kode tersebut memanggil fungsi `PlotResultsPeramalan` pada Segmen Kode Program 5.26. Fungsi tersebut merupakan proses untuk membuat grafik garis yang terdiri dari legenda target dan *output* peramalan untuk semua periode.

```
14 - function PlotResultsPeramalan(Targets, Outputs, Title)
15 -     subplot(2,2,[1 2]);
16 -     plot(Targets);
17 -     hold on;
18 -     plot(Outputs);
19 -     legend('Targets','OutputsAllForecast');
20 -     ylabel('Targets and Outputs');
21 -     grid on;
22 -     title(Title);
23 - end
```

#### Segmen Kode Program 5.26 Fungsi Plotting Data Peramalan pada Grafik Garis

*(Halaman ini sengaja dikosogkan)*

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bab ini menjelaskan mengenai hasil dari proses uji coba dan pembahasan serta analisis terhadap hasil yang diperoleh dari proses pelatihan dan pengujian jaringan GMDH dalam penyelesaian peramalan data *intermittent*.

#### **6.1. Hasil Uji Coba**

Pada sub bab ini akan dijelaskan hasil percobaan yang diperoleh dari proses uji coba program MATLAB yang telah dibuat untuk menyelesaikan permasalahan data *intermittent* menggunakan metode jaringan *Group Method of Data Handling* (GMDH).

##### **6.1.1. Lingkungan Uji Coba**

Lingkungan uji coba membahas mengenai lingkungan pengujian yang digunakan untuk implementasi tugas akhir ini meliputi perangkat keras dan perangkat lunak yang digunakan. Spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam implementasi metode jaringan GMDH ditunjukkan pada Tabel 6.1

**Tabel 6.1 Lingkungan Uji Coba**

Perangkat Keras	Spesifikasi
Laptop	ASUS X450C
Tipe Processor	Intel Core i3 Processor
Max. Memory	6 GB
Hard Drive Type	500 GB HDD
Perangkat Lunak	Fungsi
Windows 10 Pro	Sistem Operasi
MATLAB R2013a	Melakukan peramalan data
Microsoft Excel 2016	Melakukan analisa terhadap hasil uji coba

### 6.1.2. Parameter dan Skenario Uji Coba

Pada proses uji coba akan digunakan parameter yang dapat berpengaruh pada proses pelatihan jaringan GMDH. Parameter tersebut adalah parameter lag input, maksimum neuron, dan maksimum lapisan yang dijalankan. Penggunaan parameter sebagai inputan pada proses pelatihan dilakukan untuk mengetahui bagaimana kombinasi dari parameter yang dapat menghasilkan error yang minimum. Parameter dan inisiasi tersebut juga digunakan untuk mengetahui beberapa hal berikut:

- a) Mengetahui pengaruh transformasi data terhadap tingkat akurasi peramalan yang dihasilkan
- b) Mengetahui pengaruh banyaknya *lag input* dan maksimum neuron terhadap tingkat akurasi peramalan yang dihasilkan
- c) Mengetahui pengaruh banyaknya lapisan pada *hidden layer* terhadap tingkat akurasi peramalan yang dihasilkan
- d) Mengetahui pengaruh besarnya *selection pressure* terhadap tingkat akurasi peramalan yang dihasilkan

Setelah menentukan parameter yang akan digunakan pada saat uji coba, selanjutnya perlu dibuat skenario uji coba untuk mengetahui hasil pelatihan dan pengujian jaringan GMDH dengan tingkat akurasi yang optimal. Beberapa skenario tersebut antara lain sebagai berikut:

- a) Perbandingan RMSE yang diperoleh berdasarkan peramalan menggunakan data aktual tanpa transformasi dan data aktual menggunakan transformasi
- b) Perbandingan RMSE yang diperoleh berdasarkan lag input 3 periode hingga 12 periode.
- c) Perbandingan RMSE yang diperoleh berdasarkan jumlah maksimum neuron. Maksimum neuron yang disarankan yaitu sebanyak kombinasi lag input [7].
- d) Perbandingan RMSE yang diperoleh berdasarkan banyaknya lapisan pada *hidden layer*. Banyaknya maksimum lapisan pada *hidden layer* yang disarankan yaitu 10-50 [18].



- e) Perbandingan RMSE yang diperoleh berdasarkan nilai *selection pressure* ( $\alpha$ ). Nilai alpha yang yaitu berkisar  $0 \leq \alpha \leq 1$  [18].

### 6.1.3. Hasil Uji Coba Model

Pembahasan hasil uji coba akan dilakukan dengan melakukan perbandingan RMSE yang diperoleh dari seluruh percobaan. Uji coba dilakukan untuk mengetahui nilai error (RMSE) yang paling rendah dengan mencoba parameter yang berbeda-beda. RMSE ditinjau dari 2 data yaitu data pelatihan dan data pengujian. Data pengujian penting untuk ditinjau karena data tersebut digunakan sebagai validasi model dan menunjukkan apakah suatu model dapat diterapkan untuk data lain. Selain itu uji coba dilakukan terhadap data aktual transformasi dan tanpa transformasi sehingga akan diketahui model terbaik untuk mendapatkan tingkat akurasi yang optimal.

**Tabel 6.2 Uji Coba Tanpa Menggunakan Transformasi Data**

Lag Input	Maks Neuron	Maks Lapisan	$\alpha$	RMSE (pelatihan)	RMSE (pengujian)
3	3	10	0	$0,6 \times 10^{45}$	Infinite
4	6	10	0	14045	$0,4 \times 10^{37}$
5	10	10	0	11966	$0,1 \times 10^{68}$
6	15	10	0	11926	$0,8 \times 10^{94}$
7	21	10	0	11321	$0,1 \times 10^{55}$
8	28	10	0	11553	$0,2 \times 10^{94}$
9	36	10	0	9916	$0,7 \times 10^{113}$
10	45	10	0	12383	$0,5 \times 10^{66}$
11	55	10	0	11216	$0,6 \times 10^{89}$
12	66	10	0	12294	Infinite

- a) Tabel 6.2 merupakan hasil percobaan menggunakan data aktual dengan transformasi dan Tabel 6.3 tanpa transformasi. Parameter jaringan yang digunakan diantaranya: lag input 3 periode hingga 12 periode, neuron sebanyak 3 sampai 66 neuron bergantung pada

banyaknya kombinasi lag input, nilai *selection pressure* 0, dan maksimum layer sebanyak 10.

**Tabel 6.3 Uji Coba Menggunakan Transformasi Data**

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
3	3	10	0	Infinite	Infinite
4	6	10	0	17986	27329
5	10	10	0	16613	Infinite
6	15	10	0	16438	31471
7	21	10	0	17451	Infinite
8	28	10	0	16934	Infinite
9	36	10	0	14145	Infinite
10	45	10	0	16150	27797
11	55	10	0	17022	28599
12	66	10	0	20542	Infinite

Berdasarkan Tabel 6.2 dan Tabel 6.3 tersebut hasil akurasi peramalan dengan data yang ditransformasi lebih baik jika dibandingkan dengan data tanpa transformasi. Hal ini dapat terjadi karena data aktual tidak stasioner dalam ragam, sehingga perlu ditransformasi terlebih dahulu untuk menghasilkan nilai error yang minimum. Sehingga untuk proses uji coba selanjutnya dilakukan dengan menggunakan transformasi data. Dari ke-10 model dengan transformasi tersebut, model terbaik didapatkan dari skenario dengan lag input 4, maksimum neuron 6, maksimum layer 10, dan alpha 0. Hasil RMSE yang diperoleh yaitu 17986 untuk data pelatihan dan 27329 untuk pengujian.

- b) Tabel 6.4 merupakan rangkuman dari hasil percobaan menggunakan data transformasi. Untuk tabel uji coba selengkapnya dapat dilihat pada lampiran Tabel H 1. Parameter jaringan GMDH yang digunakan yaitu dengan lag input 3 periode hingga 12 periode, dan neuron sebanyak 3 sampai 66 neuron bergantung pada banyaknya kombinasi

lag input, menaikkan maksimum layer sebanyak 20-50, dan nilai alpha 0.

**Tabel 6.4 Uji Coba terhadap Penambahan Maksimum Lapisan**

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
11	55	20	0	14103	Infinite
7	21	30	0	14047	Infinite
11	55	40	0	11719	Infinite
11	55	50	0	11236	Infinite

Berdasarkan Tabel 6.4 tidak didapatkan model terbaik karena semua hasil RMSE data pengujian bernilai tak terhingga. Selain itu semakin banyak neuron dan penambahan lapisan pada jaringan, maka akan mendapatkan nilai RMSE pelatihan yang lebih baik, namun hal tersebut menyebabkan *overfitting* (Model hanya memiliki akurasi yang sangat baik untuk data pelatihan, tapi tidak untuk data pengujian) sehingga mencapai nilai RMSE pada data pengujian yang tak terhingga. Untuk uji coba selanjutnya maka dilakukan peningkatan nilai alpha hingga 0,1-1 untuk menghindari kompleksitas jaringan dan *overfitting*

- c) Tabel 6.5 merupakan rangkuman dari hasil percobaan menggunakan penambahan *selection pressure*. Untuk tabel uji coba selengkapnya dapat dilihat pada lampiran Tabel H 2 Parameter jaringan GMDH yang digunakan yaitu dengan lag input 3 periode hingga 12 periode, dan neuron sebanyak 3 sampai 66 neuron bergantung pada banyaknya kombinasi lag input, maksimum layer sebanyak 10-50, dan nilai alpha 0,1-0,5.

Berdasarkan Tabel 6.5 didapatkan model terbaik dengan nilai RMSE pelatihan 14861 dan RMSE pengujian 21721. Selain itu jika alpha yang ditentukan mendekati 1, maka semakin tinggi tekanan seleksi terhadap neuron pada jaringan. Hal ini memungkinkan mendapatkan hasil RMSE

yang minimum karena alpha yang tinggi dapat membuang neuron yang memiliki RMSE lebih besar sehingga tidak terpilih sebagai input pada penambahan lapisan selanjutnya. Untuk uji coba selanjutnya maka dilakukan peningkatan nilai alpha hingga 0,6-1 untuk menghindari kompleksitas jaringan dan *overfitting*

**Tabel 6.5 Uji Coba terhadap Penambahan Nilai Alpha 0,1-0,5**

Lag Input	Maksimum Neuron	Maksimum Lapisan	$\alpha$	RMSE (pelatihan)	RMSE (pengujian)
3	3	30	0,1	19702	26319
6	15	20	0,2	14861	21721
5	10	10	0,3	16636	29640
3	3	50	0,4	19702	26319
3	3	50	0,5	19702	26319

- d) Tabel 6.6 merupakan rangkuman dari hasil percobaan terhadap penambahan *selection pressure*. Untuk tabel uji coba selengkapnya dapat dilihat pada lampiran Tabel H 3. Parameter jaringan GMDH yang digunakan yaitu dengan lag input 3 periode hingga 12 periode, dan neuron sebanyak 3 sampai 66 neuron bergantung pada banyaknya kombinasi lag input, maksimum layer sebanyak 10-50, dan nilai alpha 0,6-1.

**Tabel 6.6 Uji Coba terhadap Penambahan Nilai Alpha 0,6-1**

Lag Input	Maksimum Neuron	Maksimum Lapisan	$\alpha$	RMSE (pelatihan)	RMSE (pengujian)
6	15	50	0,6	18129	24526
7	21	30	0,7	17761	23431
7	21	50	0,8	17761	23431
3	3	50	0,9	19843	25789
3	3	10	1	19843	25789

Berdasarkan Tabel 6.6 telah didapatkan model yang memiliki nilai akurasi lebih baik, yaitu dengan nilai RMSE pelatihan

17761 dan RMSE pengujian 23431. Model tersebut menggunakan *lag input* 7, maksimum neuron 21, maksimum lapisan 30, dan nilai *selection pressure* 0,7

#### 6.1.4. Skenario Uji Coba Terbaik

Berdasarkan hasil uji coba pada proses sebelumnya, didapatkan hasil model terbaik dari setiap skenario uji coba. Hasil terbaik dari masing-masing percobaan di tampilkan pada skenario pada Tabel 6.7

**Tabel 6.7 Nilai RMSE Model Terbaik**

Lag Input	Maksimum Neuron	Maks Lapisan	Lapisan Terbentuk	$\alpha$	RMSE (pelatihan)	RMSE (pengujian)
4	6	10	10	0	17986	27329
6	15	20	20	0,2	14861	21721
7	21	30	3	0,7	17761	23431

Dapat dilihat pada tabel bahwa hasil terbaik dari percobaan menggunakan parameter yang berbeda-beda didapatkan parameter terbaik yang menghasilkan RMSE lebih baik adalah *lag input* 6, maksimum neuron 15, maksimum lapisan 20, dan alpha 0,2. Model ke-2 ini sesuai dengan kriteria pemilihan model, yaitu memiliki nilai RMSE pada data pengujian yang lebih baik daripada model lainnya, sehingga model ke-2 akan digunakan untuk meramalkan impor beras periode mendatang. Selain itu, ketiga model tersebut akan dilakukan analisa lebih lanjut yaitu menganalisa *MDA* (*Mean Direction Accuracy*) dan *R* (*Correlation Factor*).

#### 6.2. Hasil Performa Model

Model yang akan dianalisa performanya, yaitu tiga model terbaik yang lolos uji coba. Pada sub bab ini akan dilakukan analisa performa model berdasarkan nilai RMSE, nilai MDA, dan nilai *correlation factor*.

### 6.2.1. Analisa RMSE dan MAD pada Model Jaringan GMDH

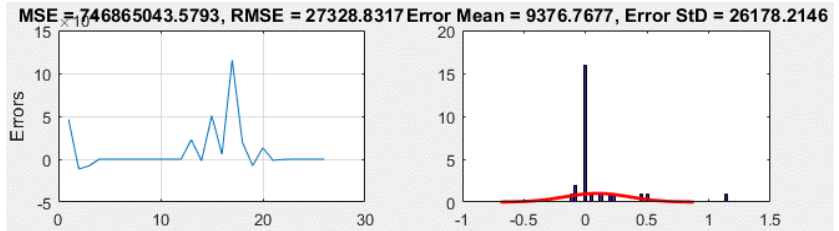
RMSE merupakan perbandingan nilai output dan target dengan melihat hasil akar dari rata-rata kuadrat dari selisih output dan target. Tabel 6.8 merupakan ketiga model terbaik yang selanjutnya akan dianalisa persebaran *error*-nya:

**Tabel 6.8 Tiga Model Terbaik**

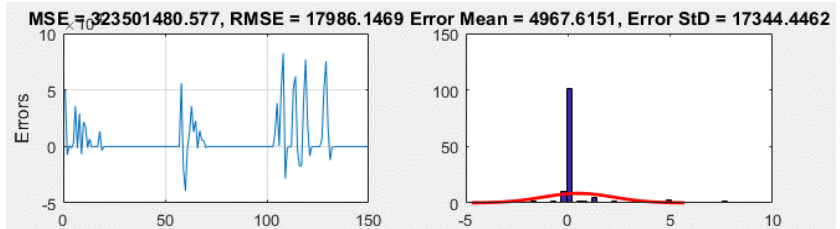
Model ke-	Lag Input	Maks Neuron	Maks Lapisan	Lapisan Terbentuk	$\alpha$
1	4	6	10	10	0
2	6	15	20	20	0,2
3	7	21	30	3	0,7

Berdasarkan Gambar 6.1 dapat dilihat persebaran error pada data pelatihan dan pengujian model ke-1. *Error* yang rendah terletak pada data aktual yang bernilai nol. Sedangkan error yang tinggi terletak pada data yang selain nol. Nilai MAD yang diperoleh yaitu 9376 untuk data pelatihan dan 4967 untuk data pengujian. Sedangkan standar deviasi error mencapai 26178 untuk data pelatihan dan 17344 untuk data pengujian. Nilai standar deviasi tersebut sangat besar, hal ini menunjukkan bahwa rentang variasi *error* pada keseluruhan data yang dihasilkan sangat besar dari nilai rata-rata (MAD).

### Train Data

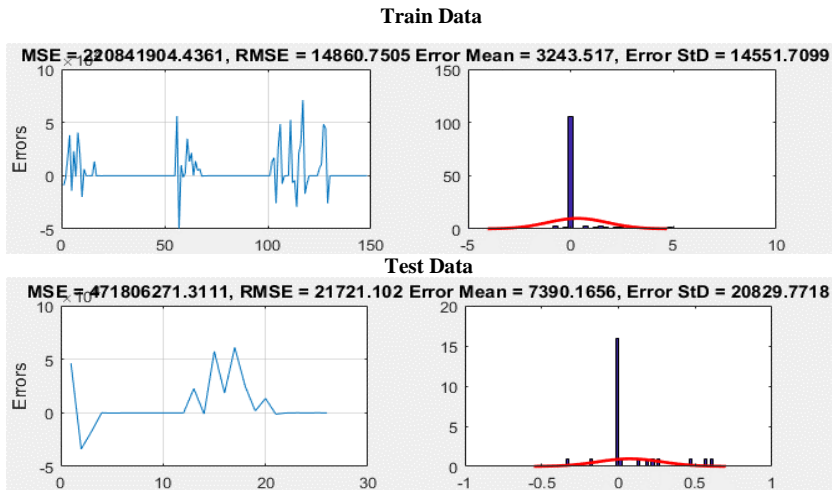


### Test Data



**Gambar 6.1 Persebaran *Error* pada Data Pelatihan dan Pengujian Model ke-1**

Berdasarkan Gambar 6.2 dapat dilihat persebaran error pada data pelatihan dan pengujian model ke-2. *Error* yang rendah terletak pada data aktual yang bernilai nol. Sedangkan error yang tinggi terletak pada data yang selain nol. Nilai MAD yang diperoleh yaitu 3243 untuk data pelatihan dan 7390 untuk data pengujian. Sedangkan standar deviasi error mencapai 14551 untuk data pelatihan dan 20829 untuk data pengujian. Nilai standar deviasi tersebut sangat besar, hal ini menunjukkan bahwa rentang variasi *error* pada keseluruhan data yang dihasilkan sangat besar dari nilai rata-rata (MAD).

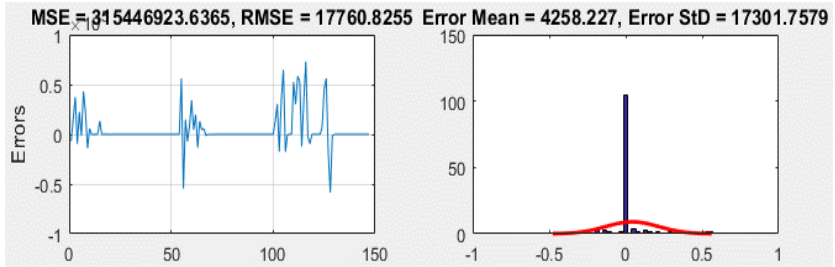


**Gambar 6.2 Persebaran *Error* pada Data Pelatihan dan Pengujian Model ke-2**

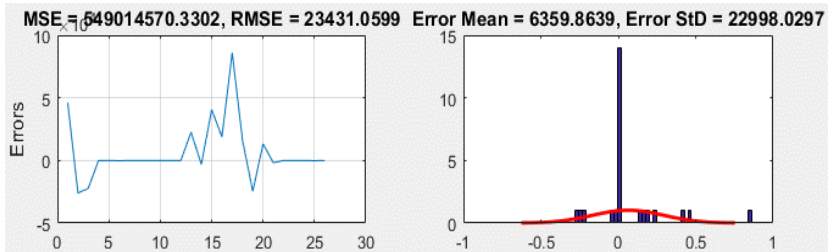
Berdasarkan Gambar 6.3 dapat dilihat persebaran error pada data pelatihan dan pengujian model ke-3. *Error* yang rendah terletak pada data aktual yang bernilai nol. Sedangkan error yang tinggi terletak pada data yang selain nol. Nilai MAD yang diperoleh yaitu 4258 untuk data pelatihan dan 6359 untuk data pengujian. Sedangkan standar deviasi error mencapai 17301 untuk data pelatihan dan 22998 untuk data pengujian. Nilai standar deviasi tersebut sangat besar, hal ini menunjukkan bahwa rentang variasi *error* pada keseluruhan data yang dihasilkan sangat besar dari nilai rata-rata (MAD).



## Train Data



## Test Data



**Gambar 6.3 Persebaran Error pada Data Pelatihan dan Pengujian Model ke-3**

Berdasarkan ketiga hasil model yang didapatkan, Tabel 6.9 merupakan perbandingan nilai RMSE ketiga model. Dapat dilihat bahwa model ke-2 adalah model terbaik diantara kedua model lainnya karena memiliki nilai RMSE pengujian yang lebih rendah. Nilai RMSE tersebut didapatkan senilai 14861 untuk RMSE pelatihan dan 21721 untuk RMSE pengujian.

**Tabel 6.9 Perbandingan Nilai RMSE**

Lag Input	Maksimum Neuron	Maks Lapisan	Lapisan Terbentuk	$\alpha$	RMSE (pelatihan)	RMSE (pengujian)
4	6	10	10	0	17986	27329
6	15	20	20	0,2	14861	21721
7	21	30	3	0,7	17761	23431

### 6.2.2. Analisa MDA pada Model Jaringan GMDH

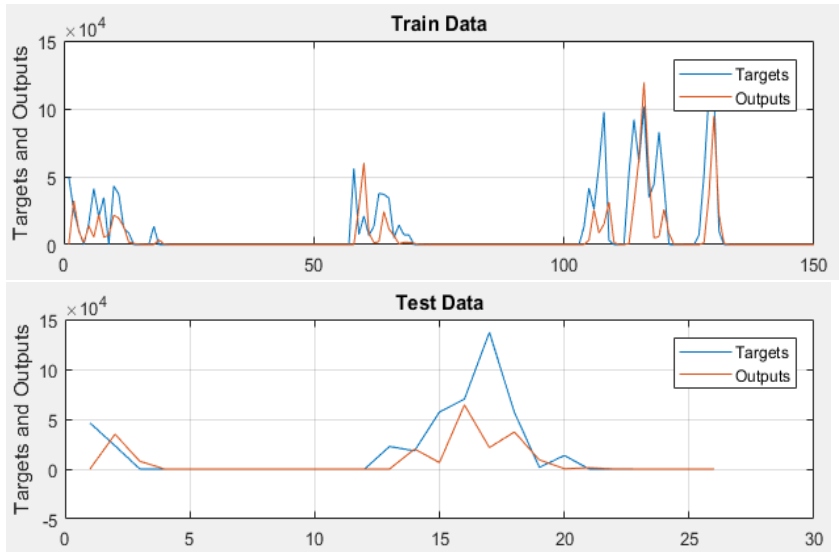
Uji MDA dilakukan untuk mengetahui apakah pola data hasil peramalan dapat mengikuti pola data aktual. Semakin besar nilai MDA, maka dapat dikatakan bahwa model dapat mengikuti pola data aktual dan sebaliknya jika nilai MDA kecil maka model tidak dapat mengikuti pola data aktual.

Tabel 6.10 merupakan hasil uji MDA terhadap 3 model. Model ke-2 memiliki nilai MDA pengujian yang lebih baik jika dibandingkan dengan kedua model lainnya dengan nilai MDA 0,798319 untuk data pelatihan dan 0,722222 untuk data pengujian.

**Tabel 6.10 Analisa Nilai Perbandingan MDA**

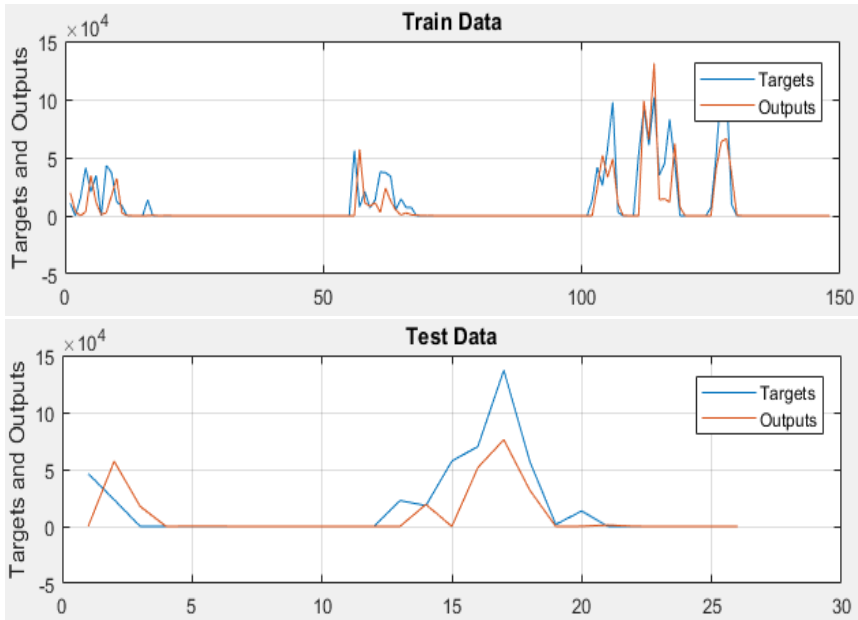
Model ke-	Lag Input	Maks Neuron	Lapisan Terbentuk	$\alpha$	MDA (pelatihan)	MDA (pengujian)
4	6	10	10	0	0,793388	0,685185
6	15	20	20	0,2	0,798319	0,722222
7	21	30	3	0,7	0,754237	0,629630

Gambar 6.4 menunjukkan pola perbandingan hasil peramalan (output) dan data aktual (target) untuk jumlah impor beras di propinsi Jawa Timur menggunakan metode jaringan GMDH model ke-1. Pada grafik garis data pelatihan ditunjukkan perbandingan pola data aktual dan data peramalan yang cukup mengikuti pola ketidakteraturan data aktual. Hal ini menandakan bahwa hasil peramalan memiliki tingkat akurasi yang cukup baik.



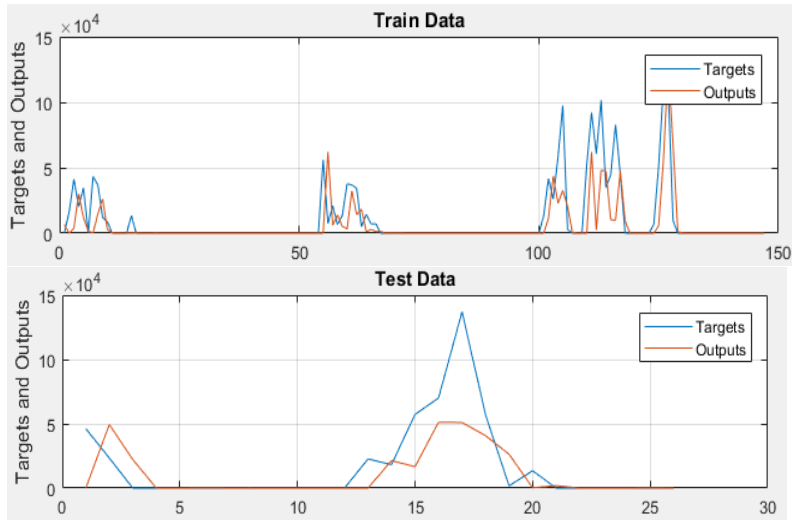
**Gambar 6.4 Perbandingan Pola Data Aktual dan Peramalan Model ke-1**

Gambar 6.5 menunjukkan pola perbandingan hasil peramalan (output) dan data aktual (target) untuk jumlah impor beras di propinsi Jawa Timur menggunakan metode jaringan GMDH model ke-2. Pada grafik garis data pelatihan ditunjukkan perbandingan pola data aktual dan data peramalan yang cukup mengikuti pola ketidakteraturan data aktual. Hal ini menandakan bahwa hasil peramalan memiliki tingkat akurasi yang cukup baik.



**Gambar 6.5 Perbandingan Pola Data Aktual dan Peramalan Model ke-2**

Gambar 6.6 menunjukkan pola perbandingan hasil peramalan (output) dan data aktual (target) untuk jumlah impor beras di propinsi Jawa Timur menggunakan metode jaringan GMDH model ke-3. Pada grafik garis data pelatihan ditunjukkan perbandingan pola data aktual dan data peramalan yang cukup mengikuti pola ketidakteraturan data aktual. Hal ini menandakan bahwa hasil peramalan memiliki tingkat akurasi yang cukup baik.

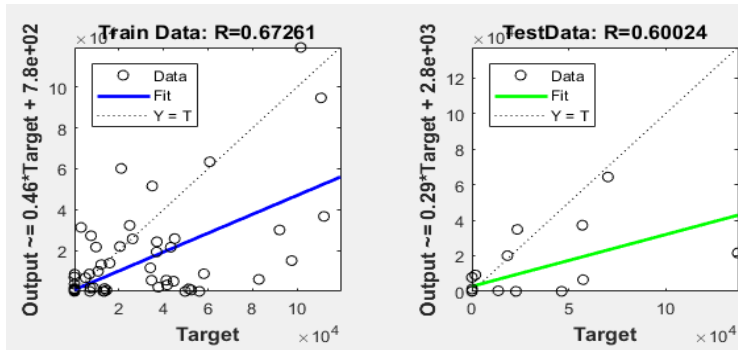


**Gambar 6.6 Perbandingan Pola Data Aktual dan Peramalan Model ke-3**

### **6.2.3. Analisa *Correlation Factor* Output dan Target pada Model Jaringan GMDH**

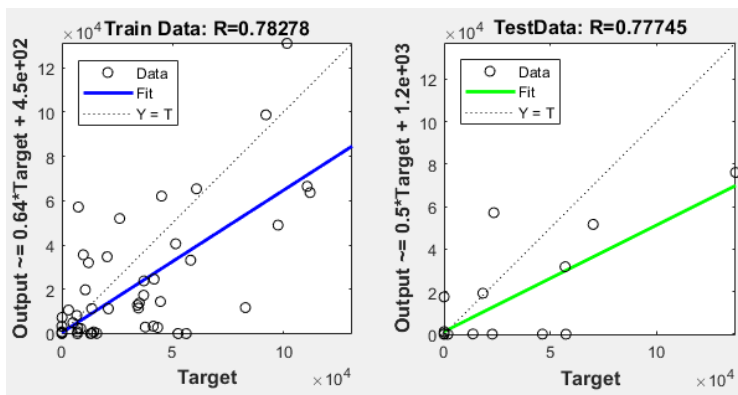
Correlation factor merupakan teknik regresi untuk mengetahui pengaruh data aktual (target) dengan data peramalan yang dihasilkan (output). Semakin besar nilai R yang dihasilkan, maka menunjukkan pengaruh yang signifikan antara data aktual dan peramalan sehingga menghasilkan nilai peramalan yang lebih baik. Nilai korelasi yang baik yaitu dengan nilai  $\leq 0,7$

Berdasarkan Gambar 6.7 diperoleh pengaruh namun tidak signifikan antara nilai target dan output GMDH data pelatihan model ke-1. Hal ini dibuktikan dengan nilai  $R = 0,67261$  pada data pelatihan  $0,60024$  pada pengujian



**Gambar 6.7 Nilai Correlation Factor pada Data Pelatihan dan Pengujian Model ke-1**

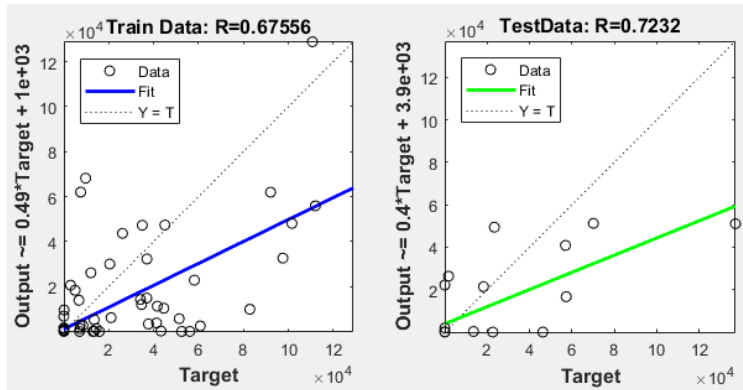
Berdasarkan Gambar 6.8 diperoleh pengaruh yang signifikan antara nilai target dan output GMDH data pelatihan model ke-2. Hal ini dibuktikan dengan nilai  $R = 0,78278$  pada data pelatihan. Pada data pengujian juga terdapat pengaruh yang tidak signifikan karena nilai  $R$  data pengujian adalah  $0,7745$



**Gambar 6.8 Nilai Correlation Factor pada Data Pelatihan dan Pengujian Model ke-2**

Berdasarkan Gambar 6.9 diperoleh pengaruh yang tidak signifikan antara nilai target dan output GMDH data pelatihan model ke-3. Hal ini dibuktikan dengan nilai  $R = 0,67556$

(mendekati 0,7) pada data pelatihan. Sedangkan pada data pengujian terdapat pengaruh yang signifikan karena nilai R data pengujian lebih dari 0,7 yaitu senilai 0,7232.



**Gambar 6.9 Nilai Correlation Factor pada Data Pelatihan dan Pengujian Model ke-3**

Tabel 6.11 merupakan perbandingan ketiga model berdasarkan nilai *correlation factor*. Model kedua dengan *lag input* 6, maksimum neuron 15, lapisan yang terbentuk sebanyak 20, dan  $\alpha$  0,2 merupakan model yang terbaik karena memperoleh nilai R paling tinggi pada proses pengujian jaringan yaitu didapatkan nilai  $R = 0,7745$

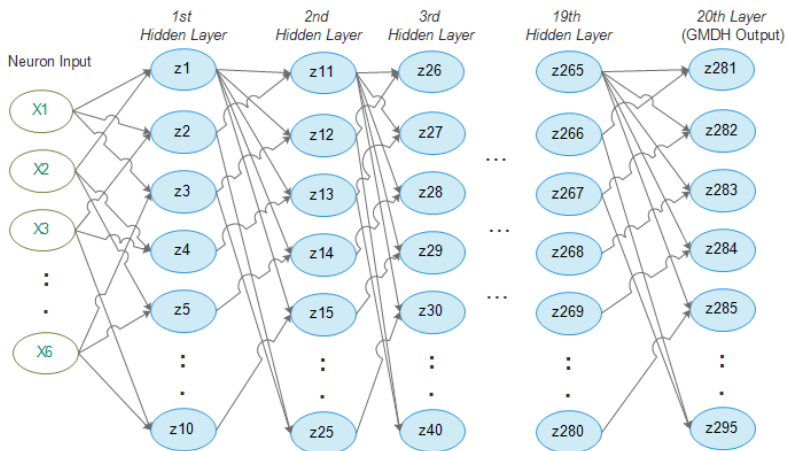
**Tabel 6.11 Pemilihan Model Terbaik Berdasarkan Nilai Korelasi Output dan Target**

Model ke-	Lag Input	Maks Neuron	Lapisan Terbentuk	$\alpha$	R (pelatihan)	R (pengujian)
4	6	10	10	0	0,6726	0,6002
6	15	20	20	0,2	0,7828	0,7745
7	21	30	3	0,7	0,6756	0,7323

### 6.3. Struktur Jaringan Model Terbaik

Pemilihan model terbaik yaitu telah didapatkan model ke-2 dengan *lag input* 6, maksimum neuron 15, maksimum lapisan 20, dan nilai *selection pressure* 0,2. Model tersebut dipilih untuk meramalkan periode selanjutnya karena sesuai dengan kriteria pemilihan model terbaik, yaitu model yang memiliki nilai RMSE minimum pada data pengujian. Proses dan hasil pemilihan model terbaik tersebut telah dijelaskan pada sub bab 6.1.3 dan 6.1.4.

Gambar 6.10 merupakan struktur jaringan GMDH yang dibentuk oleh model ke-2. Model tersebut telah terbentuk 20 lapisan dengan masing-masing memiliki nilai neuron maksimum 15. Pada setiap lapisan, terjadi penurunan minimum *error* sehingga dapat menghasilkan tingkat akurasi yang lebih baik.



**Gambar 6.10 Struktur Jaringan GMDH Model Terbaik**

Masing-masing neuron pada setiap lapisan diseleksi menggunakan indikator RMSE. Jika RMSE melebihi ambang batas maka neuron akan dibuang. Hal ini terjadi pada lapisan ke-1 pada *hidden layer*. Terdapat hanya 10 neuron yang lolos



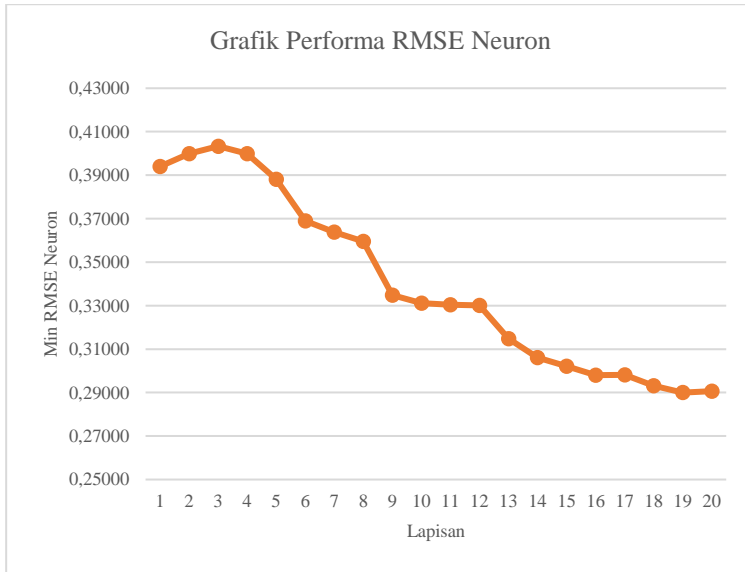
dari 15 neuron pada lapisan pertama. Pada lapisan ketiga hingga ke-19, neuron yang lolos terhadap ambang batas lebih banyak daripada jumlah maksimum neuron, sehingga algoritma akan menyeleksi neuron berdasarkan jumlah maksimum neuron layer yang telah diinisiasi. Jumlah neuron yang terbentuk dan nilai minimum RMSE neuron pada setiap lapisan dapat dilihat pada Tabel 6.12:

**Tabel 6.12 Jumlah Neuron Terbentuk dan Penurunan RMSE Neuron pada struktur jaringan GMDH Model Terbaik**

Lapisan Ke-	Jumlah Neuron Terbentuk	Minimum RMSE Neuron
1	10	0,39400
2	15	0,39998
3	15	0,40335
4	15	0,39987
5	15	0,38811
6	15	0,36901
7	15	0,36383
8	15	0,35954
9	15	0,33483
10	15	0,33108
11	15	0,33041
12	15	0,33006
13	15	0,31479
14	15	0,30612
15	15	0,30216
16	15	0,29805
17	15	0,29823
18	15	0,29310
19	15	0,28999
20	15	0,29062

Maksimum lapisan telah diinisiasi dengan nilai 20, sehingga pada lapisan ke-20 akan ditentukan neuron output pada jaringan GMDH. Output dari jaringan GMDH ini yaitu neuron dengan nilai RMSE minimum pada lapisan ke-20.

Gambar 6.11 merupakan grafik performa penurunan RMSE neuron di setiap lapisan. Pada gambar tersebut dapat dilihat bahwa setiap lapisan terdapat perbaikan RMSE, sehingga dapat menghasilkan model yang lebih baik.



**Gambar 6.11 Performa Penurunan RMSE Neuron pada Setiap Lapisan**

#### 6.4. Peramalan Periode Selanjutnya

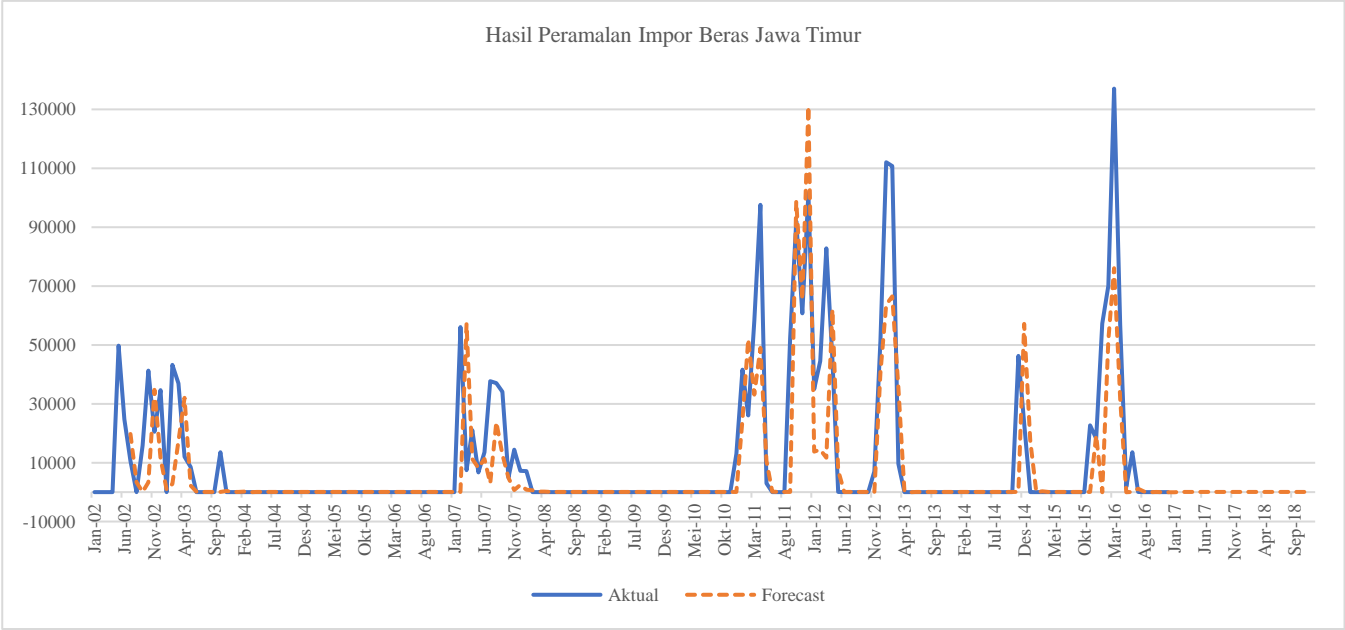
Tabel 6.13 merupakan hasil peramalan impor beras Jawa Timur untuk 2 tahun mendatang menggunakan model terbaik dari proses uji coba. Model tersebut didapatkan dengan neuron input sebanyak 6 periode, maksimum neuron 15, maksimum lapisan 20, dan alpha 0,2.

**Tabel 6.13 Hasil Peramalan 2 tahun Mendatang**

<b>Periode</b>	<b>Peramalan Jumlah Impor Beras (ton)</b>	<b>Periode</b>	<b>Peramalan Jumlah Impor Beras (ton)</b>
Jan-17	4,15000	Jan-18	14,9643
Feb-17	14,1541	Feb-18	15,2071
Mar-17	22,1829	Mar-18	14,1115
Apr-17	27,9998	Apr-18	13,7254
Mei-17	30,6667	Mei-18	13,1574
Jun-17	23,3358	Jun-18	12,6753
Jul-17	14,4244	Jul-18	12,5244
Agu-17	19,4030	Agu-18	11,9969
Sep-17	18,8465	Sep-18	11,6049
Okt-17	17,1386	Okt-18	11,2950
Nov-17	17,1138	Nov-18	10,9666
Des-17	15,0399	Des-18	10,7343

Berdasarkan Tabel 6.13 menunjukkan bahwa hasil peramalan periode 2 tahun mendatang mencapai angka maksimum 30 ton yang mana angka tersebut cenderung mendekati angka nol. Dapat dikatakan mendekati angka nol karena pada data historis sebelumnya nilai minimum impor selain angka nol yaitu senilai 40 ton, sehingga memungkinkan tidak ada permintaan impor beras pada tahun 2017-2018. Hal ini dapat terjadi karena peramalan menggunakan model dengan neuron input 6 periode dan pada 6 periode terakhir (April-Desember 2016), terdapat 5 bulan dimana tidak ada impor (data nol) .yang dapat mempengaruhi secara signifikan terhadap hasil peramalan.

Gambar 6.12 merupakan grafik perbandingan aktual dan peramalan untuk keseluruhan data jumlah impor beras Jawa Timur. Grafik ini menggambarkan data pelatihan, pengujian, dan peramalan.



**Gambar 6.12 Perbandingan Aktual dan Peramalan Importir Beras 2002-2018**

## BAB VII

### KESIMPULAN DAN SARAN

Bab ini menjelaskan kesimpulan yang didapat dari seluruh proses pengerjaan tugas akhir dan saran perbaikan untuk penelitian kedepannya untuk dapat dikembangkan dari tugas akhir ini.

#### 7.1. Kesimpulan

Kesimpulan yang didapatkan dari tugas akhir ini adalah sebagai berikut:

- 1) Model yang digunakan untuk meramalkan periode mendatang dipilih berdasarkan nilai RMSE pengujian yang terendah. Model terpilih tersebut memiliki neuron input sebanyak 6, maksimum neuron 15 pada setiap lapisan, maksimum lapisan sebanyak 20, dan dengan *selection pressure* ( $\alpha$ ) senilai 0,2. Model ini memiliki nilai RMSE 14861 untuk data pelatihan dan 21721 untuk data pengujian. Serta model dinilai dapat mengikuti pola data aktual dengan rata-rata nilai MDA (*Mean Direction Accuracy*) sebesar 0,7983 untuk data pelatihan, 0,7222 untuk data pengujian.
- 2) Semakin banyak neuron input maka RMSE jaringan pelatihan yang dihasilkan akan semakin baik, akan tetapi dapat menyebabkan RMSE pada data pengujian semakin besar. Sebaliknya jika jumlah neuron input lebih sedikit, maka akan menghasilkan RMSE yang kurang optimal pada jaringan pelatihan, namun dapat meminimalkan RMSE di data pengujian.
- 3) Peningkatan *threshold* atau *selection pressure* ( $\alpha$ ) dapat menghindari terjadinya kompleksitas jaringan dan *overfitting*, karena hal tersebut dapat menyeleksi neuron secara ketat sehingga hanya neuron dengan RMSE paling kecil yang akan dilanjutkan pada lapisan berikutnya.

Semakin besar (mendekati angka 1) nilai alpha yang ditentukan, maka akan semakin sedikit jumlah neuron dan lapisan yang dibentuk pada pelatihan jaringan. Sebaliknya, jika semakin kecil (mendekati angka 0) nilai alpha yang ditentukan, maka semakin banyak neuron pada *hidden layer* dan lapisan yang terbentuk pada jaringan.

- 4) Semakin banyak neuron pada *hidden layer* dan semakin banyak lapisan, maka dapat menghasilkan nilai RMSE yang sangat baik. Namun hal ini dapat menyebabkan *overfitting* (nilai error sangat kecil pada data pelatihan namun error sangat besar jika diterapkan pada data pengujian atau data lain).

## 7.2. Saran

Berdasarkan hasil penelitian pada tugas akhir ini, saran yang dapat penulis berikan untuk penelitian selanjutnya yaitu:

- 1) Data aktual impor beras sangat fluktuatif sehingga pada penelitian selanjutnya diharapkan dapat melakukan peramalan dengan menganalisa penyebab atau variabel-variabel yang mempengaruhi, sehingga dapat meningkatkan akurasi peramalan.
- 2) Peramalan data intermiten dapat ditambahkan fungsi transfer yang lain untuk pengembangan metode GMDH.

## DAFTAR PUSTAKA

- [1] Y. Z. Lining Bai, "Improving Inventory Management in Small Business: A Case Study," *Master Thesis in International Logistics and Supply Chain Management*, 2008.
- [2] X. L. J. K. Martin Hart, "LOGISTICS MANAGEMENT BASED ON DEMAND FORECASTING," *Research in Logistic dan Production*, vol. 3, pp. 71-80, 2013.
- [3] J. D. Croston, "Forecasting and stock control for intermittent demands.," *Operational Research Quarterly*, pp. 289-303, 1970-1977.
- [4] E. A. Silver, "Operations research in inventory management," *Operation Research*, pp. 628-624, 1981.
- [5] X.-M. Y. G. H. T. T. H. D. Prerna Mishra, "Intermittent Demand Forecast: Robustness Assesment for Group Method of Data Handling," *International Conference on Industrial Engineering and Operations Management* , pp. 1-10, 2014.
- [6] N. Kourentzes, "Intermittent Demand Forecast with Neural Network," *International Journal of Production Economics*, 2013.
- [7] R. S. Ani Shabri, "A Hybrid GMDH and Box-Jenkins Model in Time Series Forecasting," *Applied Mathematical Science*, vol. VIII, pp. 1-11, 2014.
- [8] Y. D. J. W. W. Z. Hongya Xu, "Application of GMDH to Short-Term Load Forecasting," *ResearchGate-Advances in Intelligent Systems*, pp. 27-32, 2012.
- [9] L. F. Tratar, "Improved Holt-Winter Method: A Case of Overnight Stays of Tourist in Republic of Slovenia," vol. XVI, pp. 1-13, 2013.

- [10] L. Kurzak, "IMPORTANCE OF FORECASTING IN ENTERPRISE MANAGEMENT," *Advance Logistic Systems*, vol. VI, pp. 173-182, 2012.
- [11] C. L. J. M. K. DOUGLAS C. MONTGOMERY, *Introduction to Time Series Analysis and Forecasting*, United States of America: A JOHN WILEY & SONS, INC., 2008.
- [12] B. J. B. G. McCune, *Analysis of Ecological Communities*, Gleneden Beach, Oregon , 2002.
- [13] S. J. Farlow, *Self-organizing Methods in Modeling: GMDH-type Algorithms*, New York, 1984.
- [14] Y. M. Z. ABU-KHEIL, "SYSTEM IDENTIFICATION USING GROUP METHOD OF DATA HANDLING (GMDH)," *A Thesis in Mechatronics*, 2009.
- [15] R. Samsudin dan P. Saad, "A Time Series Forecasting Model Using Group Method Of Data Handling (GMDH)," pp. 20-28.
- [16] E. I. Bueno, "Group method of data handling and neural networks applied in temperature sensors monitoring," *International Journal Nuclear Knowledge Management*, vol. V, 2011.
- [17] S. J. Farlow, "The GMDH Algorithm of Ivakhnenko," *The American Statistician*, vol. 35, 2005.
- [18] N. Ghanzanfari, "Evaluation of GMDH and MLP Networks for Prediction of Compressive Strength and Workability of Concrete," vol. 86, pp. 855-868, 2017.
- [19] P. S. A. S. Ruhaidah Samsudin, "The GMDH Model and Its Application to Forecasting of Rice Yields," *Jurnal Teknologi Maklumat*, 2008.
- [20] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, pp. 527-529, 1993.



- [21] M. H. & S. H. O. Schnader, "Evaluating predictions of change," *Journal of Business*, pp. 99-107, 1990.
- [22] J. L. Rodger, "Thirteen Ways to Look at the Correlation Coefficient," 2012.
- [23] S. Karsoliya, "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture," *International Journal of Engineering Trends and Technology*, vol. 3, no. 6, 2012.
- [24] M. Gupta, "INTELLIGENT SYSTEM USING GMDH ALGORITHM," *A Thesis*, 2010.
- [25] P. B. Josef Taušer, "EXCHANGE RATE PREDICTIONS IN INTERNATIONAL FINANCIAL MANAGEMENT BY ENHANCED GMDH ALGORITHM," *Economic Journal*, no. 3, 2011.



## **BIODATA PENULIS**

Penulis bernama lengkap Zuhriya Firda dilahirkan di Mojokerto, 27 September 1995. Penulis telah menempuh pendidikan formal sejak tahun 2000, yaitu di TK Aisiyah Bustanul Athfal Mojokerto, SDN Kedung-wonokerto Sidoarjo pada tingkat sekolah dasar, SMP Al-Rifa'ie Malang pada tingkat sekolah menengah pertama, dan MBI Amanatul Ummah Mojokerto pada tingkat sekolah menengah atas. Setelah menyelesaikan pendidikan SMA, penulis melanjutkan pendidikan tingkat Sarjana di Institut Teknologi Sepuluh Nopember, tepatnya di Departemen Sistem Informasi melalui jalur SBMPTN 2014. Selama menjadi mahasiswa, penulis aktif mengikuti kegiatan kemahasiswaan diantaranya: Staff Departemen Sosial Masyarakat HMSI (Himpunan Mahasiswa Sistem Informasi) tahun kepengurusan 2015/2016, Tim Soal BIONIX 2015-2016 yang merupakan olimpiade TI dan Bisnis tingkat Nasional untuk SMA sederajat, dan kegiatan kepedulian sosial KSR PMI Kota Surabaya serta mengikuti kegiatan kepanitiaan dan kemahasiswaan lainnya. Salah satu lomba yang pernah diikuti penulis yaitu Program Kreativitas Mahasiswa Bidang Kewirausahaan (PKM-K) yang didanai oleh DIKTI pada tahun 2015.

Penulis berfokus pada bidang minat RDIB (Rekayasa Data dan Intelegensi Bisnis) pada Departemen Sistem Informasi untuk penulisan tugas akhir dengan topik peramalan data. Jika ingin memberikan masukan dan saran mengenai tugas akhir ini, dapat menghubungi penulis melalui email [zuhriyaf@gmail.com](mailto:zuhriyaf@gmail.com).

*(Halaman ini sengaja dikosongkan)*

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN A : Data Aktual Impor Beras Jawa Timur

REALISASI PENERIMAAN BERAS LUAR NEGERI (IMPOR)															
TAHUN 2002-2014															
BULAN	TAHUN														
	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Jan	-	-	-	-	-	-	7.175	-	-	41.610	34.934	112.036	-	-	57.288
Feb	-	43.243	-	-	-	56.100	-	-	-	26.099	44.502	110.716	-	-	70.093
Mar	-	36.900	-	-	-	7.450	-	-	-	58.115	82.830	9.677	-	-	137.047
Apr	-	12.021	-	-	-	21.000	-	-	-	97.544	45.010	-	-	-	56.919
May	49.700	8.500	-	-	-	6.700	-	-	-	2.997	100	-	-	-	1.750
Jun	24.794	-	-	-	-	13.489	-	-	-	-	-	-	-	-	13.541
Jul	10.600	-	-	-	-	37.650	-	-	-	-	-	-	-	-	40
Aug	-	-	-	-	-	37.000	-	-	-	-	-	-	-	-	-
Sep	15.800	-	-	-	-	34.100	-	-	-	52.450	-	-	-	-	-
Oct	41.275	13.500	-	-	-	5.000	-	-	-	92.089	-	-	-	-	-
Nov	20.450	-	-	-	-	14.400	-	-	-	60.738	6.981	-	46.248	22.654	-
Dec	34.600	-	-	-	-	7.200	-	-	12.945	101.599	51.356	-	23.387	18.269	-
JUMLAH	197.219	114.164	-	-	-	240.089	7.175	-	12.945	533.241	265.713	232.429	69.635	40.923	336.676

**Gambar A 1 Data Aktual Impor Beras Bulog Jawa Timur 2002-2016**

A-1



## LAMPIRAN B: Neuron Input

Variables - Inputs													
Inputs													
4x176 double													
	1	2	3	4	5	6	7	8	9	10	11	12	
1	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	^
2	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	
3	1.6021	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	
4	1.6021	1.6021	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	

Variables - Inputs													
Inputs													
4x176 double													
	165	166	167	168	169	170	171	172	173	174	175	176	
1	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	1.6021	^
2	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	
3	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	
4	1.6021	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	

Gambar B 1 Neuron Input Model ke-1

Variables - Inputs													
Inputs													
7x173 double													
	1	2	3	4	5	6	7	8	9	10	11	12	
1	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	3.9315	1.6021	
2	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	3.9315	
3	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	
4	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	
5	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	
6	1.6021	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	
7	1.6021	1.6021	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	

Variables - Inputs													
Inputs													
6x174 double													
	163	164	165	166	167	168	169	170	171	172	173	174	
1	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	1.6021	
2	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	
3	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	
4	1.6021	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	
5	1.6021	1.6021	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	
6	1.6021	1.6021	1.6021	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	

**Gambar B 2 Neuron Input Model ke-2**



Variables - Inputs

Inputs Targets InverseOutputs

7x173 double

	1	2	3	4	5	6	7	8	9	10	11	12	
1	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	3.9315	1.6021	
2	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	3.9315	
3	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	
4	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	
5	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	
6	1.6021	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	
7	1.6021	1.6021	1.6021	1.6021	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	

Variables - Inputs

Inputs Targets InverseOutputs

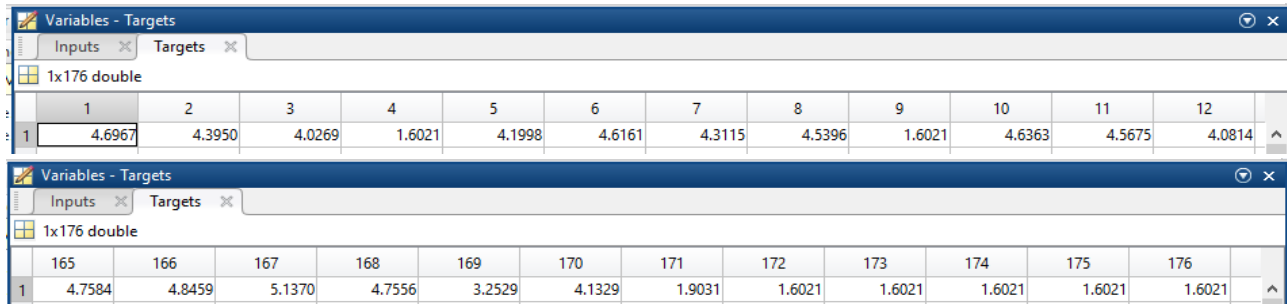
7x173 double

	163	164	165	166	167	168	169	170	171	172	173	174	
1	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	1.6021		
2	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021		
3	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021		
4	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021		
5	1.6021	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031		
6	1.6021	1.6021	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329		
7	1.6021	1.6021	1.6021	1.6021	4.3559	4.2627	4.7584	4.8459	5.1370	4.7556	3.2529		

**Gambar B 3 Neuron Input Model ke-3**



## LAMPIRAN C : Variabel Target



Variables - Targets

Inputs Targets

1x176 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	4.6967	4.3950	4.0269	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814

Variables - Targets

Inputs Targets

1x176 double

	165	166	167	168	169	170	171	172	173	174	175	176
1	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	1.6021	1.6021

Gambar C 1 Variabel Target (Yt) Model ke-1

C-2

The image displays two screenshots of a software window titled "Variables - Targets". The window has two tabs: "Inputs" and "Targets". The "Targets" tab is selected, showing a table with 12 columns and 1 row of data. The data is as follows:

	1	2	3	4	5	6	7	8	9	10	11	12
1	4.0268	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	3.9315	1.6021

The bottom screenshot shows a similar table, but with the first column labeled "1" and the first row of data. The data is as follows:

	163	164	165	166	167	168	169	170	171	172	173	174
1	4.7584	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	1.6021	1.6021

**Gambar C 2 Variabel Tarhet (Yt) Model ke-2**

The image displays two screenshots of a software window titled "Variables - Targets". The window has three tabs: "Inputs", "Targets", and "InverseOutputs". The "Targets" tab is selected in both screenshots. Below the tabs, there is a label "1x173 double".

The top screenshot shows a table with 12 columns and 1 row of data. The columns are labeled 1 through 12. The values in the row are: 1.6021, 4.1998, 4.6161, 4.3115, 4.5396, 1.6021, 4.6363, 4.5675, 4.0814, 3.9315, 1.6021, and 1.6021.

The bottom screenshot shows a table with 14 columns and 1 row of data. The columns are labeled 163 through 174. The values in the row are: 4.8459, 5.1370, 4.7556, 3.2529, 4.1329, 1.9031, 1.6021, 1.6021, 1.6021, 1.6021, 1.6021, and 1.6021.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1.6021	4.1998	4.6161	4.3115	4.5396	1.6021	4.6363	4.5675	4.0814	3.9315	1.6021	1.6021

	163	164	165	166	167	168	169	170	171	172	173	174
1	4.8459	5.1370	4.7556	3.2529	4.1329	1.9031	1.6021	1.6021	1.6021	1.6021	1.6021	1.6021

**Gambar C 3 Variabel Target (Yt) Model ke-3**



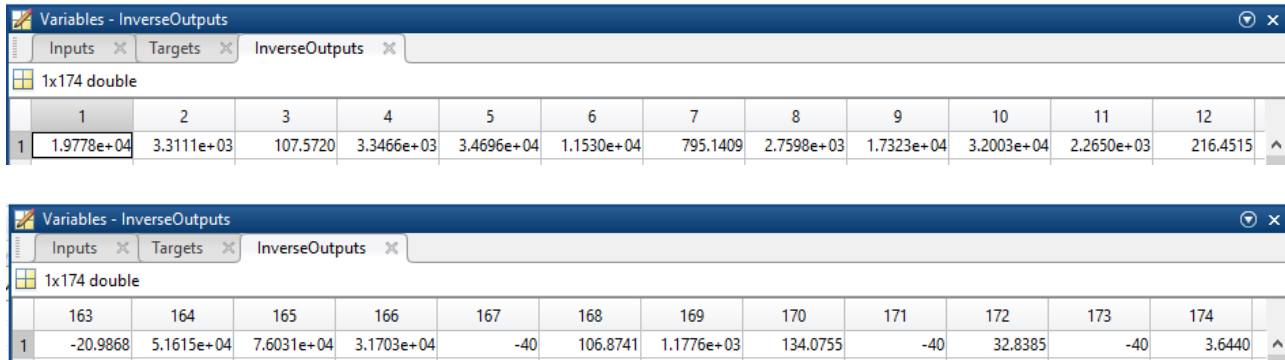
## LAMPIRAN D: Variabel Output Jaringan GMDH

Variables - InverseTestOutputs												
Inputs Targets InverseTestOutputs												
1x26 double												
	1	2	3	4	5	6	7	8	9	10	11	12
1	15.0577	3.4822e+04	7.8071e+03	-40	60.2760	-40	15.0577	15.0577	15.0577	15.0577	15.0577	15.0577

Variables - InverseOutputs												
Inputs Targets InverseTestOutputs InverseOutputs												
1x176 double												
	166	167	168	169	170	171	172	173	174	175	176	177
1e+03	6.4265e+04	2.1554e+04	3.7160e+04	9.2306e+03	308.2090	1.3384e+03	171.5118	51.6767	39.5283	10.2787	15.0577	

Gambar D 1 Variabel Output (Ft) Jaringan GMDH Model ke-1

D-2



The image displays two screenshots of a software window titled "Variables - InverseOutputs". The window has three tabs: "Inputs", "Targets", and "InverseOutputs", with the "InverseOutputs" tab selected. Below the tabs, there is a label "1x174 double" and a table of data. The first screenshot shows the first 12 columns of the table, and the second screenshot shows columns 163 through 174.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1.9778e+04	3.3111e+03	107.5720	3.3466e+03	3.4696e+04	1.1530e+04	795.1409	2.7598e+03	1.7323e+04	3.2003e+04	2.2650e+03	216.4515

	163	164	165	166	167	168	169	170	171	172	173	174
1	-20.9868	5.1615e+04	7.6031e+04	3.1703e+04	-40	106.8741	1.1776e+03	134.0755	-40	32.8385	-40	3.6440

**Gambar D 2 Variabel Output (Ft) Jaringan GMDH Model ke-2**



Variables - InverseOutputs												
Inputs Targets InverseOutputs												
1x173 double												
	1	2	3	4	5	6	7	8	9	10	11	12
1	6.7906e+03	2.4916	3.7847e+03	3.0003e+04	1.1960e+04	1.5795e+03	221.0682	1.4858e+04	2.6085e+04	2.6847e+03	240.4139	168.0751

Variables - InverseOutputs												
Inputs Targets InverseOutputs												
1x173 double												
	163	164	165	166	167	168	169	170	171	172	173	174
1	5.1234e+04	5.1047e+04	4.0817e+04	2.6321e+04	269.6425	1.9287e+03	4.5611	16.3528	21.1135	146.5749	11.8589	

**Gambar D 3 Variabel Output (Ft) Jaringan GMDH Model ke-3**



## LAMPIRAN E: Struktur Jaringan GMDH

Command Window	
New to MATLAB? See resources for <a href="#">Getting Started</a> .	
Layer 1: Neurons = 6, Min Error = 0.40919	
Layer 2: Neurons = 6, Min Error = 0.41453	
Layer 3: Neurons = 6, Min Error = 0.42996	
Layer 4: Neurons = 6, Min Error = 0.42542	
Layer 5: Neurons = 6, Min Error = 0.3918	
Layer 6: Neurons = 6, Min Error = 0.36953	
Layer 7: Neurons = 6, Min Error = 0.36006	
Layer 8: Neurons = 6, Min Error = 0.34892	
Layer 9: Neurons = 6, Min Error = 0.36807	
Layer 10: Neurons = 1, Min Error = 0.35731	

**Gambar E 1 Struktur Jaringan Model ke-1**

Command Window	
New to MATLAB? See resources for <a href="#">Getting Started</a> .	
Layer 1: Neurons = 10, Min Error = 0.394	
Layer 2: Neurons = 15, Min Error = 0.39998	
Layer 3: Neurons = 15, Min Error = 0.40335	
Layer 4: Neurons = 15, Min Error = 0.39987	
Layer 5: Neurons = 15, Min Error = 0.38811	
Layer 6: Neurons = 15, Min Error = 0.36901	
Layer 7: Neurons = 15, Min Error = 0.36383	
Layer 8: Neurons = 15, Min Error = 0.35954	
Layer 9: Neurons = 15, Min Error = 0.33483	
Layer 10: Neurons = 15, Min Error = 0.33108	
Layer 11: Neurons = 15, Min Error = 0.33041	
Layer 12: Neurons = 15, Min Error = 0.33006	
Layer 13: Neurons = 15, Min Error = 0.31479	
Layer 14: Neurons = 15, Min Error = 0.30612	
Layer 15: Neurons = 15, Min Error = 0.30216	
Layer 16: Neurons = 15, Min Error = 0.29805	
Layer 17: Neurons = 15, Min Error = 0.29823	
Layer 18: Neurons = 15, Min Error = 0.2931	
Layer 19: Neurons = 15, Min Error = 0.28999	
Layer 20: Neurons = 1, Min Error = 0.29062	

**Gambar E 2 Struktur Jaringan Model Terbaik ke-2**

Command Window	
New to MATLAB? See resources for <a href="#">Getting Started</a> .	
Layer 1: Neurons = 6, Min Error = 0.39362	
Layer 2: Neurons = 3, Min Error = 0.41164	
Layer 3: Neurons = 1, Min Error = 0.41848	

**Gambar E 3 Struktur Jaringan Model Terbaik ke-3**

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN F: Estimasi Koefisien Fungsi Transfer

Berikut merupakan estimasi koefisien pada setiap neuron dan setiap lapisan pada model ke-1

```
vars: []
c: [5.5821 -6.5733 -16.3694 12.6090 -4.2180 3.0046]
vars: [1 2]
c: [2.2264 -0.5024 -0.1149 0.2630 0.1070 -0.1489]
vars: [1 3]
c: [0.5741 -0.2978 1.0889 0.2017 -0.1491 -0.0629]
vars: [1 4]
c: [1.7239 -0.2639 0.0663 0.2018 0.0325 -0.0838]
vars: [2 3]
c: [1.5559 -0.8410 0.9953 0.2900 -0.0945 -0.1308]
vars: [2 4]
c: [1.6220 -0.5712 0.6633 0.2250 -0.0598 -0.0902]
vars: [3 4]
c: [1.2946 0.2783 0.2432 0.0744 0.0406 -0.1339]
```

Layer 1: Neurons = 6, Min Error = 0.40919

```
vars: []
c: [-0.2273 0.6306 0.5994 0.2324 0.3404 0.5305]
vars: [1 2]
c: [-0.1518 0.3715 0.7423 0.2628 0.2002 -0.4818]
vars: [1 3]
c: [-0.3933 -0.6303 1.9341 -0.0376 -0.4169 0.4051]
vars: [1 4]
c: [0.4721 1.2744 -0.6615 -0.0053 0.1898 -0.1121]
vars: [1 5]
c: [0.0156 0.9377 -0.0019 0.0445 0.0804 -0.1066]
vars: [1 6]
c: [-1.0820 1.1479 0.6979 -0.0460 -0.1586 0.0522]
vars: [2 3]
c: [-0.3983 -0.6500 1.9571 -0.0137 -0.4062 0.3698]
vars: [2 4]
c: [0.6222 1.2233 -0.7164 -0.0038 0.1851 -0.0921]
vars: [2 5]
c: [-0.1687 0.9834 0.1256 0.0229 0.0184 -0.0578]
vars: [2 6]
c: [-3.8120 1.1288 2.9230 -0.0739 -0.6498 0.1497]
vars: [3 4]
c: [0.5383 1.8803 -1.2425 -0.1784 0.1497 0.0928]
vars: [3 5]
c: [-0.3749 1.6358 -0.3115 -0.1505 -0.0266 0.1225]
vars: [3 6]
c: [-4.2853 1.5731 2.8435 -0.1396 -0.6192 0.1251]
vars: [4 5]
c: [-0.3750 -0.0803 1.3580 0.1009 -0.2064 0.0586]
vars: [4 6]
c: [-0.0054 0.6996 0.2873 -0.0025 -0.1094 0.1179]
vars: [5 6]
c: [-2.1226 1.6269 1.0569 -0.0976 -0.2223 0.0062]
```

Layer 2: Neurons = 6, Min Error = 0.41453

```
vars: []
c: [0.3260 -0.5149 1.7611 -1.3603 1.4765 1.0220]
vars: [1 2]
c: [0.0845 -2.8299 3.7792 -2.9421 -3.7533 6.7054]
vars: [1 3]
```

## F-2

```
c: [-0.1030 -1.2113 2.2997 -3.9776 -4.2926 8.2583]
vars: [1 4]
c: [0.2142 -2.2262 3.0591 0.6765 -0.0402 -0.6083]
vars: [1 5]
c: [-0.2580 -1.0323 2.2529 -0.5258 -1.0240 1.5120]
vars: [1 6]
c: [-0.1861 0.6468 0.4975 0.0063 0.2799 -0.3108]
vars: [2 3]
c: [0.0362 3.4836 -2.5054 -2.6543 -1.6650 4.3236]
vars: [2 4]
c: [-0.0241 -5.3492 6.3663 11.4051 9.6070 -21.0181]
vars: [2 5]
c: [-0.1337 -1.7047 2.8119 0.8665 0.2128 -1.0980]
vars: [2 6]
c: [-0.1608 0.9912 0.1327 -0.1939 0.0797 0.0936]
vars: [3 4]
c: [0.1953 -1.7463 2.5892 1.9169 1.2408 -3.1324]
vars: [3 5]
c: [-0.2941 -2.0347 3.2779 0.1345 -0.8560 0.6798]
vars: [3 6]
c: [-0.0903 0.1519 0.9192 -0.1255 -0.0779 0.1914]
vars: [4 5]
c: [-0.2770 -0.5067 1.7307 -0.5078 -0.8199 1.2902]
vars: [4 6]
c: [-0.2028 0.7458 0.4034 0.3110 0.4586 -0.7942]
vars: [5 6]
c: [-0.5101 2.6696 -1.2680 -0.4023 0.4531 -0.1189]
```

Layer 3: Neurons = 6, Min Error = 0.42996

```
vars: [1]
c: [0.0799 0.4702 0.2273 0.3240 0.2229 0.1784]
vars: [1 2]
c: [0.1737 -4.5006 5.3624 1.2020 -0.4528 -0.7260]
vars: [1 3]
c: [0.4352 1.2325 -0.5718 2.2647 2.5308 -4.7400]
vars: [1 4]
c: [0.0187 -2.6614 3.6315 2.3771 1.4836 -3.8573]
vars: [1 5]
c: [0.1589 -2.8550 3.7286 2.0898 1.0393 -3.1092]
vars: [1 6]
c: [0.3205 -7.0386 7.7905 1.6303 -0.3723 -1.2173]
vars: [2 3]
c: [0.5503 3.6499 -3.0810 0.8699 1.6857 -2.4846]
vars: [2 4]
c: [-0.1114 -1.4826 2.5625 1.1852 0.6329 -1.8320]
vars: [2 5]
c: [0.0172 -2.2337 3.2191 0.4459 -0.4008 -0.0425]
vars: [2 6]
c: [0.2770 -1.7703 2.5574 2.4169 1.8040 -4.1877]
vars: [3 4]
c: [0.1801 -1.5949 2.4489 0.9963 0.4018 -1.3742]
vars: [3 5]
c: [0.3839 -4.7177 5.4217 -1.6482 -3.4940 5.1947]
vars: [3 6]
c: [0.5278 -6.0991 6.6881 -1.8122 -3.8447 5.7283]
vars: [4 5]
c: [0.5950 -3.8078 4.3647 -2.0234 -3.6240 5.7223]
vars: [4 6]
c: [-0.3176 0.7867 0.4770 -1.3492 -1.2292 2.5337]
vars: [5 6]
c: [-0.4039 6.0547 -4.7253 -7.9534 -5.6081 13.5069]
```

Layer 4: Neurons = 6, Min Error = 0.42542

```
vars: []
c: [-0.3837 0.1232 0.9497 0.3124 -1.5208 1.6694]
vars: [1 2]
c: [0.2064 -1.7600 2.5942 2.9421 2.1400 -5.0557]
vars: [1 3]
c: [0.4103 -3.4135 4.0735 3.8357 2.5782 -6.3576]
vars: [1 4]
c: [0.5290 -6.6612 7.2063 7.7320 5.5323 -13.1886]
vars: [1 5]
c: [-0.2317 -9.7218 10.9219 -9.3748 -11.5291 20.8768]
vars: [1 6]
c: [-0.1941 -0.7646 1.9283 -0.4477 -0.9714 1.3911]
vars: [2 3]
c: [0.1623 -2.8467 3.7139 2.8001 1.5624 -4.3406]
vars: [2 4]
c: [-0.0123 -2.3337 3.3448 0.8716 -0.1522 -0.7217]
vars: [2 5]
c: [0.4601 -5.1867 5.8438 -1.5524 -3.0004 4.6135]
vars: [2 6]
c: [0.1585 -0.6527 1.5289 0.3498 0.0615 -0.3909]
vars: [3 4]
c: [-0.2072 1.3453 -0.1833 -2.7049 -2.5481 5.2272]
vars: [3 5]
c: [-0.1817 0.1141 1.0266 -0.7975 -0.8697 1.6456]
vars: [3 6]
c: [-0.0516 0.0397 0.9953 0.1997 0.0799 -0.2850]
vars: [4 5]
c: [-0.4070 -0.5207 1.8502 -2.4178 -2.5416 4.9068]
vars: [4 6]
c: [-0.2685 0.4653 0.7504 -0.5923 -0.6564 1.2134]
vars: [5 6]
c: [-1.8000 2.4700 -0.0142 -5.0045 -5.3257 10.0891]
```

Layer 5: Neurons = 6, Min Error = 0.3918

```
vars: []
c: [0.0674 0.1397 0.2201 0.1674 0.2489 0.2600]
vars: [1 2]
c: [-0.3434 -1.0059 2.2795 -3.6349 -4.2827 7.8767]
vars: [1 3]
c: [0.1411 0.9140 -0.0594 2.2032 2.4491 -4.6285]
vars: [1 4]
c: [0.0100 1.7606 -0.7679 0.6323 0.9391 -1.5721]
vars: [1 5]
c: [0.0641 1.3345 -0.3855 0.4405 0.6277 -1.0611]
vars: [1 6]
c: [0.0087 1.3733 -0.3798 -0.4288 -0.3244 0.7558]
vars: [2 3]
c: [0.9208 2.3041 -2.1385 7.4653 9.4529 -16.7723]
vars: [2 4]
c: [0.2713 4.4050 -3.6125 -0.3158 0.6801 -0.3299]
vars: [2 5]
c: [0.0279 1.6151 -0.6391 0.7681 1.0768 -1.8425]
vars: [2 6]
c: [-0.4227 4.0682 -2.7194 -3.5062 -3.0578 6.5104]
vars: [3 4]
c: [0.1167 1.3942 -0.5236 6.4572 6.4463 -12.8877]
vars: [3 5]
c: [-0.1708 -2.1364 3.2647 5.1389 4.4486 -9.6152]
vars: [3 6]
c: [-0.2236 -1.6344 2.8143 0.1247 -0.6392 0.4838]
vars: [4 5]
```

## F-4

```

c: [-0.1124 1.2830 -0.1970 0.3194 0.5236 -0.8576]
vars: [4 6]
c: [-0.6321 1.7267 -0.2353 -3.4952 -3.6582 7.0763]
vars: [5 6]
c: [-0.2953 -0.9232 2.1413 -5.0549 -5.9637 10.9895]

Layer 6: Neurons = 6, Min Error = 0.36953
vars: []
c: [0.6809 0.4198 -0.2549 0.1799 -0.2259 0.0022]
vars: [1 2]
c: [-0.5550 2.7863 -1.3592 -1.2686 -0.9543 2.1579]
vars: [1 3]
c: [-0.4175 3.3361 -2.0304 -0.5326 0.0601 0.4269]
vars: [1 4]
c: [-0.4898 2.8895 -1.5181 -1.1385 -0.7749 1.8586]
vars: [1 5]
c: [-0.3735 2.6147 -1.3349 -0.9243 -0.6090 1.4935]
vars: [1 6]
c: [-0.3020 1.2705 -0.0697 -1.4751 -1.7082 3.1664]
vars: [2 3]
c: [0.2668 3.4447 -2.6975 8.1383 8.8511 -16.9498]
vars: [2 4]
c: [-0.0769 -3.3304 4.3930 6.0289 5.1124 -11.1531]
vars: [2 5]
c: [-0.0816 -1.5153 2.5998 16.3146 16.4476 -32.7891]
vars: [2 6]
c: [-0.3147 -2.8783 4.1308 -3.2179 -4.5637 7.7461]
vars: [3 4]
c: [0.0325 -1.6643 2.6057 6.2605 5.7868 -12.0392]
vars: [3 5]
c: [-0.0972 -0.2079 1.2633 3.6550 3.7043 -7.3714]
vars: [3 6]
c: [-0.5206 -1.1189 2.5275 -1.4846 -2.2304 3.6507]
vars: [4 5]
c: [0.0701 3.8913 -2.9476 -0.7537 0.0300 0.7338]
vars: [4 6]
c: [-0.2422 -1.5340 2.7271 -2.1947 -3.0420 5.2093]
vars: [5 6]
c: [-0.2532 -2.1509 3.3584 -2.1609 -2.9822 5.1122]

Layer 7: Neurons = 6, Min Error = 0.36006
vars: []
c: [1.3502 1.7246 -1.5377 0.7520 -1.4186 1.1873]
vars: [1 2]
c: [0.5440 2.2550 -1.6800 0.7007 1.3597 -1.9930]
vars: [1 3]
c: [-0.5898 0.7502 0.7244 -0.8176 -0.9380 1.6799]
vars: [1 4]
c: [0.6457 0.6693 -0.2017 1.1236 1.5146 -2.5509]
vars: [1 5]
c: [-0.7427 0.8700 0.7315 -1.2377 -1.3772 2.5199]
vars: [1 6]
c: [-0.1787 -0.3428 1.4739 0.0142 -0.2320 0.1988]
vars: [2 3]
c: [0.1257 -0.0122 0.9087 0.0834 0.0763 -0.1420]
vars: [2 4]
c: [-0.0821 0.3714 0.6921 0.2029 -0.0200 -0.1938]
vars: [2 5]
c: [0.1158 -0.0771 0.9857 -0.0708 -0.1233 0.2102]
vars: [2 6]
c: [-0.0137 -0.9937 2.0065 0.3255 -0.0137 -0.3144]
vars: [3 4]

```



```

c: [0.3751 -0.5037 1.2120 0.0908 -0.4924 0.4493]
vars: [3 5]
c: [0.1083 3.8114 -2.9204 11.0983 11.6052 -22.6857]
vars: [3 6]
c: [-0.1032 0.5265 0.5513 -1.8933 -1.9498 3.8357]
vars: [4 5]
c: [0.2253 0.8042 0.0229 -0.4361 -0.0600 0.5248]
vars: [4 6]
c: [-0.0830 -0.5918 1.6677 -0.0987 -0.2813 0.3674]
vars: [5 6]
c: [-0.1444 0.8798 0.2372 -2.6117 -2.6383 5.2355]

```

Layer 8: Neurons = 6, Min Error = 0.34892

```

vars: []
c: [0.8723 0.1312 -0.4912 0.0651 -0.6859 -0.1791]
vars: [1 2]
c: [-0.3249 0.0732 1.1796 -0.2538 -0.4982 0.7124]
vars: [1 3]
c: [-0.0623 -0.4755 1.5256 -0.1763 -0.4365 0.6062]
vars: [1 4]
c: [-0.2161 0.0374 1.1329 -0.5667 -0.7726 1.3136]
vars: [1 5]
c: [0.1138 -2.2273 3.1341 0.0912 -0.5594 0.4848]
vars: [1 6]
c: [-0.3307 -0.0692 1.3258 -0.1859 -0.4802 0.6260]
vars: [2 3]
c: [-0.3495 1.0910 0.2023 -2.3748 -1.8949 4.2196]
vars: [2 4]
c: [0.2918 -0.9714 1.7351 2.9397 2.6112 -5.5142]
vars: [2 5]
c: [-0.5368 1.2525 0.1726 -1.1376 -0.7812 1.8518]
vars: [2 6]
c: [0.0947 1.4015 -0.4911 8.5462 8.6112 -17.1432]
vars: [3 4]
c: [-0.3595 0.5907 0.7046 -3.0086 -2.9640 5.9258]
vars: [3 5]
c: [-0.0793 1.4856 -0.4232 -0.2870 -0.0624 0.3401]
vars: [3 6]
c: [-0.3764 -0.0101 1.3243 -1.5319 -2.1197 3.5980]
vars: [4 5]
c: [-0.3466 0.5209 0.7544 -1.7347 -1.5927 3.2855]
vars: [4 6]
c: [0.0728 0.8377 0.1005 0.8841 0.9226 -1.7976]
vars: [5 6]
c: [-0.5112 0.0928 1.3102 -0.5656 -0.9204 1.4225]

```

Layer 9: Neurons = 6, Min Error = 0.36807

```

vars: []
c: [0.4872 0.0578 0.0348 0.1280 -0.2213 -0.2415]
vars: [1 2]
c: [-0.2700 1.0779 0.1570 -3.1512 -3.0146 6.1263]
vars: [1 3]
c: [-0.2359 3.4103 -2.1992 -7.0152 -6.1590 13.1406]
vars: [1 4]
c: [0.1151 0.8264 0.0743 -1.1779 -1.0602 2.2581]
vars: [1 5]
c: [-0.0552 1.9463 -0.8984 -2.5387 -1.8263 4.3571]
vars: [1 6]
c: [-0.0589 -1.3970 2.4430 2.1496 1.8595 -4.0170]
vars: [2 3]
c: [-0.0248 1.2297 -0.2017 -0.9528 -0.5301 1.4779]
vars: [2 4]

```

## F-6

```
c: [-0.0332 0.8896 0.1401 -1.3301 -1.2134 2.5409]
vars: [2 5]
c: [-0.0658 2.1367 -1.0723 -1.7271 -1.0617 2.7778]
vars: [2 6]
c: [-0.2379 0.1364 1.0688 -2.1726 -2.2814 4.4198]
vars: [3 4]
c: [0.0132 -0.5415 1.5464 -3.2152 -3.4096 6.6278]
vars: [3 5]
c: [-0.1106 -0.1685 1.2719 -2.0156 -2.3186 4.3169]
vars: [3 6]
c: [-0.2124 -2.4081 3.5997 -7.5665 -8.3612 15.8989]
vars: [4 5]
c: [-0.1010 -3.0977 4.1777 1.5460 0.7169 -2.2775]
vars: [4 6]
c: [0.1805 -0.1878 1.0299 -2.4321 -2.5891 5.0550]
vars: [5 6]
c: [-0.0494 -1.4889 2.5471 -5.2879 -5.9905 11.2687]
```

Layer 10: Neurons = 1, Min Error = 0.35731

Berikut merupakan estimasi koefisien pada setiap neuron dan setiap lapisan pada model ke-2

```
vars: [6 13]
c: [0.1442 -0.4307 1.3228 -0.8856 -1.1788 2.0830]
vars: [6 14]
c: [-0.0345 0.7247 0.3030 -0.3228 -0.3010 0.6199]
vars: [6 15]
c: [0.2787 1.5512 -0.7800 -5.8416 -5.5697 11.4532]
vars: [7 8]
c: [0.1096 0.0670 0.8419 -1.5993 -1.5950 3.2115]
vars: [7 9]
c: [-0.0947 1.3695 -0.2959 -1.0053 -0.7539 1.7494]
vars: [7 10]
c: [0.0223 2.0182 -1.0347 -0.3625 -0.1174 0.4826]
vars: [7 11]
c: [-0.0797 1.4997 -0.4348 -1.0781 -0.7994 1.8680]
vars: [7 12]
c: [0.0313 -0.8096 1.7786 1.6078 1.5754 -3.1791]
vars: [7 13]
c: [0.0834 -1.3348 2.2689 1.0765 0.4345 -1.5011]
vars: [7 14]
c: [-0.0931 1.5266 -0.4512 -1.2054 -0.8828 2.0775]
vars: [7 15]
c: [-0.0112 -0.4469 1.4566 -1.8833 -2.2481 4.1300]
vars: [8 9]
c: [-0.0383 0.4349 0.5959 -0.3573 -0.4473 0.8002]
vars: [8 10]
c: [0.0744 -1.4521 2.3903 -2.4648 -3.2089 5.6861]
vars: [8 11]
c: [-0.0286 0.3999 0.6243 -0.3081 -0.4400 0.7444]
vars: [8 12]
c: [0.0691 0.6712 0.2804 -1.2062 -1.0762 2.2907]
vars: [8 13]
c: [0.1546 -0.4114 1.2941 -1.0841 -1.3799 2.4846]
vars: [8 14]
c: [-0.0635 0.6776 0.3743 -0.4164 -0.4125 0.8208]
vars: [8 15]
c: [0.1332 1.2322 -0.3459 -2.1633 -2.0768 4.2622]
vars: [9 10]
```

```

c: [-0.1260 -0.5807 1.6777 -0.8852 -1.2500 2.1222]
vars: [9 11]
c: [-0.0054 0.7272 0.2768 0.0249 0.1642 -0.1898]
vars: [9 12]
c: [0.0228 -0.1988 1.1806 0.0555 -0.0344 -0.0179]
vars: [9 13]
c: [0.1258 -2.0630 2.9745 0.1463 -0.4929 0.3601]
vars: [9 14]
c: [0.0752 3.8302 -2.8902 -3.5037 -2.4464 5.9608]
vars: [9 15]
c: [-0.1161 0.2430 0.8469 -0.4652 -0.5730 1.0251]
vars: [10 11]
c: [-0.1301 1.7925 -0.6885 -1.5609 -1.1432 2.6888]
vars: [10 12]
c: [0.2291 -0.1207 0.9345 0.8240 1.0255 -1.8195]
vars: [10 13]
c: [0.0291 -0.3309 1.3100 -0.6447 -0.8635 1.5124]
vars: [10 14]
c: [-0.1079 1.7961 -0.7098 -1.5373 -1.1035 2.6290]
vars: [10 15]
c: [0.0228 0.5836 0.3967 -2.0182 -1.9610 3.9839]
vars: [11 12]
c: [0.1366 -0.7450 1.6378 0.4380 0.3020 -0.7233]
vars: [11 13]
c: [0.1301 -2.0539 2.9635 -0.2688 -0.9529 1.2359]
vars: [11 14]
c: [0.0561 7.6340 -6.6780 -24.5632 -22.3404 46.9122]
vars: [11 15]
c: [-0.0566 -0.4530 1.4980 -0.0807 -0.3305 0.4043]
vars: [12 13]
c: [0.1488 -0.1434 1.0368 -1.0136 -1.2287 2.2608]
vars: [12 14]
c: [0.1098 1.8106 -0.8964 0.2455 0.4592 -0.6914]
vars: [12 15]
c: [0.0164 1.8792 -0.8951 1.8848 1.9162 -3.8000]
vars: [13 14]
c: [0.1643 3.0259 -2.1432 -0.8767 -0.1686 1.0640]
vars: [13 15]
c: [-0.0060 1.5379 -0.5297 -0.7321 -0.3844 1.1153]
vars: [14 15]
c: [-0.0704 -0.3014 1.3571 -0.2155 -0.4600 0.6673]

```

Layer 14: Neurons = 15, Min Error = 0.30612

```

vars: []
c: [1.6497 0.1218 -1.5728 -0.7508 2.0231 -2.1612]
vars: [1 2]
c: [-0.1811 1.9424 -0.7945 5.1332 5.3180 -10.4780]
vars: [1 3]
c: [-0.2214 3.2810 -2.1138 -1.6280 -0.9653 2.5682]
vars: [1 4]
c: [-0.2089 3.0941 -1.9268 2.7500 3.1967 -5.9755]
vars: [1 5]
c: [-0.0497 -2.8276 3.8698 2.2476 1.4679 -3.7237]
vars: [1 6]
c: [-0.0031 0.2177 0.7842 -0.0806 -0.2039 0.2845]
vars: [1 7]
c: [0.0091 -1.4248 2.4163 1.6496 1.0145 -2.6641]
vars: [1 8]
c: [-0.1942 2.3075 -1.1485 5.6565 5.8762 -11.5619]
vars: [1 9]
c: [-0.1659 0.7136 0.4195 -1.1552 -1.1832 2.3175]
vars: [1 10]

```

```

c: [0.0070 -1.4952 2.4866 1.7918 1.1465 -2.9384]
vars: [1 11]
c: [0.0149 0.0718 0.9167 -0.1395 -0.2989 0.4407]
vars: [1 12]
c: [-0.1699 2.8219 -1.6860 1.0855 1.4623 -2.5713]
vars: [1 13]
c: [0.0468 0.2365 0.7295 -0.3978 -0.4659 0.8700]
vars: [1 14]
c: [0.0436 -1.3371 2.2928 2.1650 1.9639 -4.1215]
vars: [1 15]
c: [-0.1735 5.2934 -4.1572 0.6380 1.6197 -2.2801]
vars: [2 3]
c: [-0.2816 5.6686 -4.4562 -4.3291 -3.1004 7.3981]
vars: [2 4]
c: [0.0017 8.1715 -7.1688 -121.9759 -119.8257 241.8026]
vars: [2 5]
c: [-0.0962 -2.1468 3.2242 2.0408 1.5639 -3.6188]
vars: [2 6]
c: [-0.0192 0.1425 0.8719 -0.0040 -0.1400 0.1418]
vars: [2 7]
c: [-0.1255 -3.7698 4.8591 4.7417 3.2596 -8.0176]
vars: [2 8]
c: [0.0035 -2.0510 3.0493 101.5104 98.5240 -200.0367]
vars: [2 9]
c: [-0.1533 1.5402 -0.4137 -1.9385 -1.7631 3.6816]
vars: [2 10]
c: [-0.1525 -4.7921 5.8962 5.7948 3.9692 -9.7827]
vars: [2 11]
c: [-0.0072 -0.0338 1.0402 -0.1193 -0.3173 0.4357]
vars: [2 12]
c: [-0.1249 5.6037 -4.5050 0.9316 1.9359 -2.8840]
vars: [2 13]
c: [0.0250 0.2288 0.7552 -0.4429 -0.5198 0.9659]
vars: [2 14]
c: [-0.0045 -1.2954 2.2879 2.1936 2.1032 -4.2957]
vars: [2 15]
c: [-0.0164 3.1394 -2.1239 2.5816 3.1850 -5.7703]
vars: [3 4]
c: [-0.2104 -4.1688 5.3241 -3.7576 -4.8992 8.6352]
vars: [3 5]
c: [-0.1962 -1.3233 2.4806 -0.0732 -0.5247 0.5722]
vars: [3 6]
c: [0.0303 0.0543 0.9206 -0.2491 -0.3986 0.6528]
vars: [3 7]
c: [-0.2610 -2.5143 3.7122 3.2543 2.3118 -5.6001]
vars: [3 8]
c: [-0.2555 -4.6642 5.8504 -6.1782 -7.3626 13.5159]
vars: [3 9]
c: [-0.0671 1.5047 -0.4496 -1.9074 -1.7222 3.6229]
vars: [3 10]
c: [-0.2866 -2.7603 3.9739 3.5611 2.5530 -6.1505]
vars: [3 11]
c: [0.0164 -0.0872 1.0742 -0.2482 -0.4426 0.6938]
vars: [3 12]
c: [0.0293 -0.5544 1.5152 -7.0821 -7.3197 14.4144]
vars: [3 13]
c: [0.0687 0.1100 0.8383 -0.4275 -0.5407 0.9779]
vars: [3 14]
c: [-0.0779 -0.5731 1.6331 0.3763 0.1385 -0.5247]
vars: [3 15]
c: [-0.1270 -1.5652 2.6642 0.5062 -0.0217 -0.5007]
vars: [4 5]

```

```

c: [-0.1301 -2.3678 3.4725 2.3517 1.8302 -4.2008]
vars: [4 6]
c: [-0.0263 0.1308 0.8896 -0.0650 -0.2115 0.2734]
vars: [4 7]
c: [-0.1320 -2.2327 3.3303 2.7033 1.7999 -4.5200]
vars: [4 8]
c: [-0.0602 -0.9991 2.0463 -15.7554 -16.4936 32.2417]
vars: [4 9]
c: [-0.1528 1.2223 -0.0967 -1.6895 -1.6071 3.2767]
vars: [4 10]
c: [-0.1638 -2.8998 4.0171 3.5430 2.4110 -5.9741]
vars: [4 11]
c: [-0.0156 -0.0591 1.0726 -0.1576 -0.3704 0.5260]
vars: [4 12]
c: [-0.0550 3.4305 -2.3871 0.4147 0.9661 -1.3880]
vars: [4 13]
c: [0.0054 0.1941 0.8053 -0.3798 -0.4706 0.8510]
vars: [4 14]
c: [-0.0539 -2.0687 3.0960 3.2522 2.9775 -6.2344]
vars: [4 15]
c: [0.0307 5.1791 -4.1976 3.6142 4.6545 -8.2677]
vars: [5 6]
c: [-0.0222 0.8537 0.1652 -0.5091 -0.4831 0.9898]
vars: [5 7]
c: [-0.0675 0.4939 0.5660 -1.1440 -1.2422 2.3768]
vars: [5 8]
c: [-0.0688 2.7669 -1.7119 1.2751 1.6152 -2.9005]
vars: [5 9]
c: [-0.0968 -0.1185 1.2037 -1.4126 -1.7327 3.1311]
vars: [5 10]
c: [-0.0648 0.5932 0.4633 -0.8025 -0.8760 1.6694]
vars: [5 11]
c: [-0.0118 0.5922 0.4193 -0.5007 -0.5379 1.0375]
vars: [5 12]
c: [-0.1171 2.8540 -1.7604 1.3139 1.6670 -2.9978]
vars: [5 13]
c: [0.0034 0.5743 0.4247 -0.3993 -0.4011 0.8014]
vars: [5 14]
c: [-0.1443 3.3864 -2.2706 -5.4322 -4.6755 10.0897]
vars: [5 15]
c: [-0.0688 3.9911 -2.9359 2.1309 2.8305 -4.9717]
vars: [6 7]
c: [-0.0085 -0.5317 1.5365 -0.0688 -0.2862 0.3549]
vars: [6 8]
c: [-0.0204 0.8292 0.1863 -0.2146 -0.1014 0.3138]
vars: [6 9]
c: [-0.0403 0.5811 0.4552 -0.7324 -0.6859 1.4131]
vars: [6 10]
c: [0.0154 -0.8072 1.7940 -0.0622 -0.3311 0.3961]
vars: [6 11]
c: [0.0972 -5.1491 6.0816 -17.3129 -18.9277 36.2521]
vars: [6 12]
c: [-0.0101 0.9132 0.0955 -0.4025 -0.2565 0.6580]
vars: [6 13]
c: [-0.0311 1.1734 -0.1462 -2.0823 -1.8676 3.9464]
vars: [6 14]
c: [0.0041 0.5386 0.4595 -0.3792 -0.2740 0.6539]
vars: [6 15]
c: [0.0035 0.8642 0.1330 -0.2847 -0.1464 0.4319]
vars: [7 8]
c: [-0.1378 4.2832 -3.1874 2.8890 4.0893 -6.9945]
vars: [7 9]

```

## F-10

```

c: [-0.1397 0.9928 0.1266 -1.0161 -0.8927 1.8893]
vars: [7 10]
c: [-0.1527 -2.7106 3.8145 86.9465 87.9219 -174.8844]
vars: [7 11]
c: [0.0283 0.5545 0.4207 -0.5124 -0.5300 1.0476]
vars: [7 12]
c: [-0.1519 2.9855 -1.8804 2.6113 3.1499 -5.7782]
vars: [7 13]
c: [0.1018 0.2459 0.6748 -1.0486 -1.0878 2.1514]
vars: [7 14]
c: [-0.0590 0.8887 0.1595 -0.5032 -0.3101 0.8058]
vars: [7 15]
c: [-0.0364 3.0717 -2.0476 1.7397 2.5124 -4.2574]
vars: [8 9]
c: [-0.1406 2.0151 -0.8971 -2.2921 -2.0019 4.2752]
vars: [8 10]
c: [-0.1642 -4.0321 5.1417 5.0158 3.5499 -8.5839]
vars: [8 11]
c: [-0.0132 0.0065 1.0047 -0.1726 -0.3495 0.5206]
vars: [8 12]
c: [-0.2103 14.0372 -12.8735 -1.2079 1.7394 -0.5579]
vars: [8 13]
c: [0.0144 0.2769 0.7158 -0.4634 -0.5194 0.9846]
vars: [8 14]
c: [-0.0092 -0.7928 1.7913 1.5229 1.5260 -3.0488]
vars: [8 15]
c: [-0.0192 1.6829 -0.6675 0.2186 0.5818 -0.8031]
vars: [9 10]
c: [-0.1324 0.3289 0.7833 -0.7622 -0.8217 1.5656]
vars: [9 11]
c: [-0.0331 0.2284 0.8045 -0.7195 -0.8201 1.5349]
vars: [9 12]
c: [-0.0912 0.4290 0.6461 -1.0811 -1.0949 2.1642]
vars: [9 13]
c: [-0.0053 0.5574 0.4493 -0.4533 -0.4552 0.9080]
vars: [9 14]
c: [-0.1758 1.4182 -0.2717 -1.0729 -0.7433 1.7923]
vars: [9 15]
c: [-0.2099 -0.0785 1.2489 -1.6946 -1.7445 3.4121]
vars: [10 11]
c: [0.0267 0.8540 0.1226 -0.4979 -0.4583 0.9610]
vars: [10 12]
c: [-0.1776 3.6063 -2.4888 3.3132 3.9784 -7.3100]
vars: [10 13]
c: [0.0999 0.3237 0.5977 -1.2011 -1.2343 2.4505]
vars: [10 14]
c: [-0.0622 0.8207 0.2294 -0.3799 -0.2051 0.5770]
vars: [10 15]
c: [-0.0384 3.2379 -2.2152 1.9684 2.7677 -4.7411]
vars: [11 12]
c: [-0.0080 1.1193 -0.1106 -0.5052 -0.2829 0.7869]
vars: [11 13]
c: [-0.1306 3.2630 -2.1562 -4.1479 -3.5190 7.6514]
vars: [11 14]
c: [0.0192 0.7247 0.2628 -0.4998 -0.3481 0.8504]
vars: [11 15]
c: [0.0108 1.0215 -0.0291 -0.3443 -0.1550 0.5009]
vars: [12 13]
c: [-0.0042 0.2667 0.7411 -0.4403 -0.4960 0.9354]
vars: [12 14]
c: [-0.0728 -1.2342 2.2832 1.6949 1.5007 -3.2043]
vars: [12 15]

```

```

c: [-0.0402 -0.7444 1.7741 -0.5445 -0.7699 1.3101]
vars: [13 14]
c: [0.0182 0.6496 0.3387 -0.4178 -0.3374 0.7578]
vars: [13 15]
c: [0.0459 0.7285 0.2398 -0.5594 -0.4922 1.0575]
vars: [14 15]
c: [0.0019 4.0418 -3.0619 3.7696 4.3583 -8.1248]

Layer 15: Neurons = 15, Min Error = 0.30216
vars: []
c: [0.1428 0.3161 -0.1135 0.4078 -0.2841 0.0030]
vars: [1 2]
c: [0.2071 -0.5570 1.3838 -9.4919 -9.7779 19.3033]
vars: [1 3]
c: [0.0212 -0.0074 0.9914 -6.7258 -6.6740 13.4032]
vars: [1 4]
c: [0.1552 -0.8228 1.7018 -39.3354 -38.9471 78.3056]
vars: [1 5]
c: [0.1440 -2.5186 3.4121 -0.3876 -1.7969 2.2015]
vars: [1 6]
c: [0.0740 0.0090 0.9348 -0.6225 -0.7289 1.3610]
vars: [1 7]
c: [0.0931 -0.1213 1.0411 -1.4014 -1.5276 2.9454]
vars: [1 8]
c: [0.1323 -2.8691 3.7683 0.8585 -0.7745 -0.0677]
vars: [1 9]
c: [0.1303 -0.6723 1.5752 -0.7316 -1.1598 1.9071]
vars: [1 10]
c: [0.0984 -3.2718 4.2009 -22.4266 -22.9592 45.3997]
vars: [1 11]
c: [0.0800 -0.2972 1.2345 0.0018 -0.2882 0.2968]
vars: [1 12]
c: [-0.0337 -0.7372 1.7647 11.2037 10.7584 -21.9698]
vars: [1 13]
c: [0.1187 -0.6433 1.5538 -0.4426 -0.8732 1.3303]
vars: [1 14]
c: [0.0696 -0.0834 1.0286 -0.0753 -0.2167 0.3012]
vars: [1 15]
c: [0.1232 -0.5283 1.4354 -0.5840 -0.9503 1.5496]
vars: [2 3]
c: [0.0807 -0.2621 1.1914 -4.4542 -4.6584 9.1271]
vars: [2 4]
c: [0.1494 5.5494 -4.6700 -2.4091 -1.2943 3.7245]
vars: [2 5]
c: [0.0134 -0.6611 1.6532 -1.0278 -1.4106 2.4402]
vars: [2 6]
c: [0.0887 -1.3036 2.2309 1.2966 0.5846 -1.8694]
vars: [2 7]
c: [0.0551 0.0198 0.9354 -0.2300 -0.3661 0.6040]
vars: [2 8]
c: [-0.0229 -0.8239 1.8429 0.2845 -0.1625 -0.1253]
vars: [2 9]
c: [0.0278 -0.7035 1.6880 -1.2062 -1.6727 2.8815]
vars: [2 10]
c: [0.0417 1.7815 -0.8120 1.5017 1.6804 -3.1781]
vars: [2 11]
c: [0.0491 -0.6543 1.6143 -7.1196 -7.4368 14.5663]
vars: [2 12]
c: [0.0376 -0.4401 1.4102 -3.2716 -3.5872 6.8656]
vars: [2 13]
c: [0.0240 -0.6808 1.6669 -0.8320 -1.2923 2.1266]
vars: [2 14]

```

```

c: [0.0934 -1.5235 2.4439 1.7158 0.9341 -2.6370]
vars: [2 15]
c: [0.0358 -0.6250 1.6024 -1.0728 -1.5163 2.5928]
vars: [3 4]
c: [-0.0193 7.3665 -6.3505 18.5310 20.7846 -39.3197]
vars: [3 5]
c: [0.0444 -1.3913 2.3595 -0.1543 -0.9233 1.0826]
vars: [3 6]
c: [0.0617 -0.1242 1.0753 -0.0157 -0.1646 0.1885]
vars: [3 7]
c: [0.1322 -0.3363 1.2236 -2.5149 -2.7531 5.2911]
vars: [3 8]
c: [0.0250 -1.3648 2.3455 0.6318 -0.1376 -0.4911]
vars: [3 9]
c: [0.0887 -0.6233 1.5563 -0.2823 -0.7012 0.9943]
vars: [3 10]
c: [-0.0106 2.6193 -1.6114 8.4997 9.0755 -17.5775]
vars: [3 11]
c: [-0.0234 0.2008 0.8181 5.8317 5.6577 -11.4947]
vars: [3 12]
c: [0.0535 -0.6264 1.5803 4.7667 4.3747 -9.1348]
vars: [3 13]
c: [0.0817 -0.6772 1.6147 0.0463 -0.4109 0.3747]
vars: [3 14]
c: [0.0607 -0.2282 1.1785 0.3339 0.1503 -0.4759]
vars: [3 15]
c: [0.0855 -0.5702 1.5041 0.0762 -0.3175 0.2521]
vars: [4 5]
c: [0.0028 -2.2026 3.2024 0.6337 -0.5187 -0.1152]
vars: [4 6]
c: [0.0903 -0.5606 1.4880 0.5590 0.2871 -0.8342]
vars: [4 7]
c: [0.1425 -0.9191 1.7999 -1.1584 -1.4761 2.6570]
vars: [4 8]
c: [-0.0155 -2.0392 3.0504 1.3054 0.2456 -1.5528]
vars: [4 9]
c: [0.0968 -1.0011 1.9286 -0.2677 -0.8393 1.1187]
vars: [4 10]
c: [0.1091 -2.6843 3.5987 17.1590 16.0877 -33.2332]
vars: [4 11]
c: [0.0491 -0.8392 1.8011 5.5592 5.1927 -10.7481]
vars: [4 12]
c: [0.0448 -1.5053 2.4677 6.9890 6.4617 -13.4464]
vars: [4 13]
c: [0.0875 -1.0856 2.0191 0.2169 -0.3887 0.1824]
vars: [4 14]
c: [0.0944 -0.6414 1.5639 0.7984 0.4979 -1.2836]
vars: [4 15]
c: [0.0984 -0.9897 1.9143 0.1688 -0.3776 0.2210]
vars: [5 6]
c: [0.0279 1.1573 -0.1879 -4.4960 -4.2570 8.7613]
vars: [5 7]
c: [0.0039 1.1416 -0.1483 -1.3398 -1.1649 2.5083]
vars: [5 8]
c: [0.0217 -9.6896 10.6624 47.2289 43.7256 -90.9498]
vars: [5 9]
c: [-0.1580 4.5935 -3.4670 2.6283 3.3796 -6.0296]
vars: [5 10]
c: [-0.0379 2.5199 -1.4878 -0.6841 0.1139 0.5648]
vars: [5 11]
c: [-0.0866 1.8443 -0.7782 4.9712 5.1979 -10.1822]
vars: [5 12]

```



```

c: [0.0237 1.1512 -0.1670 -1.0070 -0.8790 1.8887]
vars: [5 13]
c: [-0.1880 3.5481 -2.3945 6.0630 6.5038 -12.5942]
vars: [5 14]
c: [0.1254 -0.5587 1.4574 -1.6005 -1.7725 3.3911]
vars: [5 15]
c: [-0.2134 5.8861 -4.7170 1.6598 2.7053 -4.3934]
vars: [6 7]
c: [-0.0150 0.8382 0.1760 -1.0094 -0.8947 1.9021]
vars: [6 8]
c: [-0.1217 -5.1706 6.2561 -6.0052 -7.3934 13.3893]
vars: [6 9]
c: [0.0503 1.3198 -0.3723 -12.8955 -12.6676 25.5774]
vars: [6 10]
c: [0.0572 1.4682 -0.5151 0.4256 0.6715 -1.0894]
vars: [6 11]
c: [-0.0129 1.0326 -0.0239 0.6547 0.6594 -1.3157]
vars: [6 12]
c: [0.0622 0.7233 0.2328 -1.6033 -1.6215 3.2326]
vars: [6 13]
c: [0.0997 1.0967 -0.1945 -16.8269 -16.5627 33.4134]
vars: [6 14]
c: [0.0218 -4.5883 5.5662 24.1522 22.5390 -46.6880]
vars: [6 15]
c: [0.0943 2.2188 -1.3099 -15.5393 -15.0814 30.6425]
vars: [7 8]
c: [0.0139 -0.1626 1.1504 -0.9435 -1.1174 2.0648]
vars: [7 9]
c: [0.0114 0.0999 0.8893 -1.4310 -1.5328 2.9679]
vars: [7 10]
c: [0.1203 1.8849 -0.9868 -1.4623 -1.1266 2.6083]
vars: [7 11]
c: [0.0576 0.8531 0.0933 -2.4857 -2.3338 4.8328]
vars: [7 12]
c: [0.0808 0.4336 0.4930 -2.1978 -2.1707 4.3853]
vars: [7 13]
c: [0.0270 0.2123 0.7644 -1.6333 -1.6932 3.3328]
vars: [7 14]
c: [-0.0405 -0.6886 1.7182 0.8127 0.4543 -1.2720]
vars: [7 15]
c: [0.0234 0.2230 0.7564 -1.5230 -1.5900 3.1189]
vars: [8 9]
c: [-0.1280 7.3145 -6.2089 1.5199 2.8671 -4.4054]
vars: [8 10]
c: [-0.0669 2.6861 -1.6340 0.1681 1.0263 -1.2031]
vars: [8 11]
c: [-0.0831 2.2225 -1.1615 3.2964 3.6278 -6.9355]
vars: [8 12]
c: [0.0125 1.1566 -0.1641 -0.5848 -0.4416 1.0277]
vars: [8 13]
c: [-0.1612 7.1943 -6.0623 2.0686 3.3438 -5.4354]
vars: [8 14]
c: [-0.0268 1.9795 -0.9673 -3.7892 -3.3741 7.1647]
vars: [8 15]
c: [-0.1367 8.4738 -7.3636 -0.9792 0.6550 0.3060]
vars: [9 10]
c: [0.0622 1.7031 -0.7484 -1.0026 -0.5344 1.5443]
vars: [9 11]
c: [0.0291 1.5074 -0.5338 1.3080 1.6352 -2.9387]
vars: [9 12]
c: [0.0789 1.4008 -0.4566 -1.4205 -1.1437 2.5731]
vars: [9 13]

```

```

c: [0.0227 -11.5137 12.4989 74.1677 70.6117 -144.7783]
vars: [9 14]
c: [0.1029 -1.2154 2.1318 -3.8297 -4.1334 7.9789]
vars: [9 15]
c: [0.0298 12.1043 -11.1264 -228.2064 -224.9414 453.1530]
vars: [10 11]
c: [0.0090 -0.7319 1.7273 5.9612 5.7144 -11.6780]
vars: [10 12]
c: [0.0169 -0.8919 1.8789 0.5740 0.1691 -0.7412]
vars: [10 13]
c: [0.0517 -0.8152 1.7772 -0.1438 -0.6392 0.7891]
vars: [10 14]
c: [0.0626 -0.6243 1.5716 0.8733 0.5918 -1.4565]
vars: [10 15]
c: [0.0622 -0.7100 1.6629 -0.0587 -0.4870 0.5534]
vars: [11 12]
c: [0.0770 1.4250 -0.4765 -5.2173 -5.0691 10.2952]
vars: [11 13]
c: [0.0306 -0.7699 1.7405 2.5520 2.1350 -4.6821]
vars: [11 14]
c: [0.0114 -0.4017 1.3892 1.1705 1.0518 -2.2205]
vars: [11 15]
c: [0.0257 -0.6719 1.6446 3.3245 3.0161 -6.3358]
vars: [12 13]
c: [0.0803 -0.4895 1.4317 -0.9442 -1.2763 2.2299]
vars: [12 14]
c: [0.0522 0.2067 0.7559 -0.8705 -0.8754 1.7526]
vars: [12 15]
c: [0.0809 -0.4042 1.3469 -1.1753 -1.4489 2.6334]
vars: [13 14]
c: [0.1150 -1.1140 2.0188 -5.2662 -5.5608 10.8456]
vars: [13 15]
c: [-0.0162 6.5614 -5.5469 29.0355 30.8695 -59.9082]
vars: [14 15]
c: [0.1218 2.1661 -1.2664 -5.6392 -5.3204 10.9790]

Layer 16: Neurons = 15, Min Error = 0.29805
vars: []
c: [1.2488 -1.3453 0.2182 -0.8068 -0.3844 -1.3315]
vars: [1 2]
c: [0.0824 -1.0233 1.9510 -15.3508 -16.0477 31.4150]
vars: [1 3]
c: [-0.0654 3.3633 -2.3059 7.0651 7.4845 -14.5617]
vars: [1 4]
c: [-0.0150 0.4256 0.5857 0.2519 0.2307 -0.4844]
vars: [1 5]
c: [-0.0723 -1.0255 2.0796 1.6217 1.1266 -2.7576]
vars: [1 6]
c: [-0.0116 -1.1941 2.1989 -2.6277 -3.0181 5.6471]
vars: [1 7]
c: [0.0052 0.6871 0.3080 0.0504 0.0577 -0.1072]
vars: [1 8]
c: [0.1127 9.7730 -8.8546 1.6922 3.6477 -5.3291]
vars: [1 9]
c: [-0.0433 -0.3931 1.4225 1.2735 1.0700 -2.3482]
vars: [1 10]
c: [0.0044 -0.2150 1.2148 -0.7700 -0.9079 1.6779]
vars: [1 11]
c: [-0.0216 -0.7083 1.7264 0.3370 -0.1688 -0.1713]
vars: [1 12]
c: [-0.0307 -0.5932 1.6197 0.0272 -0.3515 0.3199]
vars: [1 13]

```

```

c: [-0.0204 -0.6190 1.6357 0.3776 -0.0395 -0.3410]
vars: [1 14]
c: [0.0213 7.1507 -6.1534 -17.0679 -14.9183 31.9887]
vars: [1 15]
c: [-0.0649 -0.7979 1.8470 1.0692 0.6420 -1.7196]
vars: [2 3]
c: [-0.1405 6.1144 -5.0027 0.8145 2.0000 -2.8335]
vars: [2 4]
c: [-0.0588 1.1518 -0.1072 -1.1497 -1.0045 2.1478]
vars: [2 5]
c: [-0.0358 0.2692 0.7598 -0.5039 -0.6073 1.1069]
vars: [2 6]
c: [0.0228 0.8384 0.1353 -3.3971 -3.1549 6.5603]
vars: [2 7]
c: [-0.0147 1.4954 -0.4863 -0.4069 -0.1922 0.5984]
vars: [2 8]
c: [-0.0226 2.9375 -1.9248 -1.1882 -0.6118 1.7999]
vars: [2 9]
c: [-0.0358 0.3557 0.6719 -0.1070 -0.1571 0.2599]
vars: [2 10]
c: [-0.0124 0.0230 0.9910 -1.2379 -1.3298 2.5659]
vars: [2 11]
c: [-0.0279 0.0612 0.9644 -0.7292 -0.9175 1.6426]
vars: [2 12]
c: [-0.0362 -0.3592 1.3910 -0.1573 -0.4661 0.6181]
vars: [2 13]
c: [-0.0188 -0.0868 1.1048 -0.4205 -0.6357 1.0532]
vars: [2 14]
c: [-0.0148 3.7381 -2.7305 -4.7157 -3.7499 8.4669]
vars: [2 15]
c: [-0.0306 -5.2138e-04 1.0259 -0.3269 -0.4970 0.8200]
vars: [3 4]
c: [0.0364 -1.0630 2.0356 -2.8206 -3.2039 6.0295]
vars: [3 5]
c: [-0.0515 -2.5002 3.5320 3.6850 2.6763 -6.3666]
vars: [3 6]
c: [-0.0676 -2.4987 3.5492 -5.3406 -6.1775 11.5119]
vars: [3 7]
c: [-0.0287 0.1373 0.8816 3.2562 3.0971 -6.3574]
vars: [3 8]
c: [0.0019 0.2637 0.7347 0.1725 0.0540 -0.2263]
vars: [3 9]
c: [-0.0021 -1.0941 2.0858 2.7414 2.4710 -5.2105]
vars: [3 10]
c: [0.0054 -0.3097 1.3080 -0.5450 -0.6948 1.2401]
vars: [3 11]
c: [0.0255 -0.7181 1.6960 1.0123 0.6390 -1.6477]
vars: [3 12]
c: [0.0184 -0.6840 1.6691 0.4027 0.0753 -0.4756]
vars: [3 13]
c: [0.0369 -0.9308 1.8995 1.1162 0.6709 -1.7821]
vars: [3 14]
c: [0.0841 3.4275 -2.4837 -10.5050 -10.0224 20.5390]
vars: [3 15]
c: [-0.0512 -2.2249 3.2584 2.8969 1.9191 -4.8217]
vars: [4 5]
c: [-0.2029 -2.4663 3.6195 3.3382 2.3846 -5.7484]
vars: [4 6]
c: [0.0565 0.2138 0.7336 -4.8793 -4.9162 9.8084]
vars: [4 7]
c: [0.2720 2.7546 -1.9746 -7.8841 -7.1470 15.0732]
vars: [4 8]

```

```

c: [0.1175 1.7207 -0.8191 -4.3598 -3.9916 8.3711]
vars: [4 9]
c: [-0.0896 -1.9022 2.9604 3.5448 2.9564 -6.5102]
vars: [4 10]
c: [-0.0148 -0.3072 1.3192 -0.1204 -0.2954 0.4140]
vars: [4 11]
c: [-0.0421 -0.6282 1.6618 0.3047 -0.1162 -0.1941]
vars: [4 12]
c: [-0.0530 -0.6393 1.6807 0.3176 -0.0482 -0.2762]
vars: [4 13]
c: [-0.0538 -1.0537 2.0951 0.9588 0.4070 -1.3728]
vars: [4 14]
c: [0.1838 3.3202 -2.4541 -8.3626 -7.6143 16.0000]
vars: [4 15]
c: [-0.1774 -2.1497 3.2846 2.7140 1.8513 -4.5880]
vars: [5 6]
c: [-0.0264 2.9421 -1.9324 4.8892 5.2739 -10.1646]
vars: [5 7]
c: [-0.0159 1.6860 -0.6788 1.3457 1.6555 -3.0022]
vars: [5 8]
c: [-0.0188 2.3393 -1.3330 2.7016 3.0766 -5.7794]
vars: [5 9]
c: [0.0171 3.1313 -2.1542 -14.5963 -14.1763 28.7806]
vars: [5 10]
c: [-0.0032 0.0849 0.9161 -3.0854 -3.0183 6.1050]
vars: [5 11]
c: [0.2519 -5.3019 6.0954 -27.4391 -29.2993 56.7792]
vars: [5 12]
c: [0.2010 -6.3252 7.2066 -11.4667 -13.5412 25.0305]
vars: [5 13]
c: [0.4599 -12.6027 13.2580 -37.4588 -41.3966 78.9186]
vars: [5 14]
c: [0.0034 1.7645 -0.7658 -0.3490 0.1309 0.2182]
vars: [5 15]
c: [0.2201 -17.1194 18.0016 -127.4380 -131.7825 259.2365]
vars: [6 7]
c: [-0.0594 3.2604 -2.2079 -5.0043 -3.9790 8.9748]
vars: [6 8]
c: [-0.0789 3.5265 -2.4646 -2.5730 -1.7085 4.2719]
vars: [6 9]
c: [-0.0301 0.2540 0.7691 0.1630 0.2326 -0.3995]
vars: [6 10]
c: [0.0340 -0.1542 1.1287 -1.9126 -1.9669 3.8849]
vars: [6 11]
c: [0.0137 0.1791 0.8105 -0.8739 -0.8864 1.7625]
vars: [6 12]
c: [0.0434 -0.2875 1.2564 -0.6482 -0.7917 1.4454]
vars: [6 13]
c: [0.0306 -0.1004 1.0770 -0.2518 -0.3041 0.5598]
vars: [6 14]
c: [0.0640 2.5542 -1.6089 -3.9384 -3.5197 7.4707]
vars: [6 15]
c: [0.0055 -1.6506 2.6364 4.0393 3.6639 -7.7008]
vars: [7 8]
c: [0.0364 -0.1497 1.1153 3.2263 2.8655 -6.0860]
vars: [7 9]
c: [0.0274 0.0652 0.9128 -0.0483 -0.0992 0.1512]
vars: [7 10]
c: [0.1227 -0.8675 1.7753 -1.5748 -1.8467 3.4378]
vars: [7 11]
c: [0.1053 -0.5118 1.4327 -1.0565 -1.3161 2.3861]
vars: [7 12]

```

```

c: [0.0850 -0.6639 1.6045 -0.7928 -1.0864 1.8894]
vars: [7 13]
c: [0.1128 -0.5883 1.5049 -0.9403 -1.2081 2.1624]
vars: [7 14]
c: [0.0703 2.4439 -1.5069 -2.9804 -2.5593 5.5531]
vars: [7 15]
c: [0.0120 -0.5680 1.5573 0.5052 0.1820 -0.6853]
vars: [8 9]
c: [0.0017 -0.2876 1.2837 1.1160 1.0282 -2.1441]
vars: [8 10]
c: [0.0410 -0.2895 1.2573 -0.2804 -0.4272 0.7134]
vars: [8 11]
c: [0.0415 -0.2230 1.1899 0.1137 -0.0355 -0.0727]
vars: [8 12]
c: [0.0512 -0.4128 1.3741 -0.2384 -0.4488 0.6940]
vars: [8 13]
c: [0.0530 -0.3093 1.2677 0.0905 -0.0730 -0.0107]
vars: [8 14]
c: [0.0867 1.7929 -0.8710 -4.7473 -4.5362 9.3022]
vars: [8 15]
c: [-0.0167 -0.8128 1.8203 1.6346 1.3301 -2.9658]
vars: [9 10]
c: [0.0015 -0.7985 1.7974 -2.1307 -2.3844 4.5157]
vars: [9 11]
c: [0.1830 -5.8464 6.6984 -33.0066 -35.0546 68.0893]
vars: [9 12]
c: [0.1403 -7.2925 8.2349 -15.2688 -17.5632 32.8400]
vars: [9 13]
c: [0.3057 -14.0588 14.8515 -73.6407 -77.8383 151.5148]
vars: [9 14]
c: [0.0306 1.9525 -0.9763 -0.2140 0.4673 -0.2495]
vars: [9 15]
c: [0.0485 -7.3457 8.3008 -15.7186 -17.2150 32.9457]
vars: [10 11]
c: [-0.0318 3.1843 -2.1615 -2.7153 -2.3392 5.0518]
vars: [10 12]
c: [0.0614 -1.7356 2.6957 -2.8847 -3.6122 6.5038]
vars: [10 13]
c: [0.0747 -1.9312 2.8737 -0.6151 -1.2819 1.9064]
vars: [10 14]
c: [0.0902 2.4582 -1.5177 -2.1377 -1.5020 3.6491]
vars: [10 15]
c: [0.0095 -1.4625 2.4553 -0.8266 -1.3761 2.2042]
vars: [11 12]
c: [0.0099 -3.3771 4.4016 -10.0232 -11.0570 21.0749]
vars: [11 13]
c: [0.2132 -6.1302 6.9822 -74.3586 -75.2253 149.6075]
vars: [11 14]
c: [0.0820 2.6888 -1.7449 -1.4534 -0.4710 1.9329]
vars: [11 15]
c: [0.0437 -3.9955 4.9555 -12.2419 -13.0221 25.2744]
vars: [12 13]
c: [-0.0318 5.7635 -4.7136 -17.6345 -16.1002 33.7263]
vars: [12 14]
c: [0.0770 2.8465 -1.8941 -1.3136 -0.3119 1.6325]
vars: [12 15]
c: [0.0657 5.6849 -4.7279 -12.7525 -11.0459 23.8116]
vars: [13 14]
c: [0.0704 2.6826 -1.7282 -1.2942 -0.3293 1.6301]
vars: [13 15]
c: [0.0781 2.3875 -1.4640 -24.8897 -23.8589 48.7685]
vars: [14 15]

```

```

c: [0.0165 -0.7381 1.7285 -0.2131 -0.6544 0.8690]

Layer 17: Neurons = 15, Min Error = 0.29823
vars: []
c: [0.3233 -0.0647 0.0873 0.2903 0.9651 -0.7245]
vars: [1 2]
c: [-0.0241 0.4507 0.5661 -11.3008 -11.4397 22.7407]
vars: [1 3]
c: [0.0260 -0.1769 1.1555 5.9878 6.2222 -12.2072]
vars: [1 4]
c: [0.0991 -2.0863 3.0079 -30.9269 -31.6930 62.6359]
vars: [1 5]
c: [-0.1357 1.8191 -0.7031 27.3407 27.0898 -54.4549]
vars: [1 6]
c: [-0.2450 2.5097 -1.3134 29.7011 29.4305 -59.1722]
vars: [1 7]
c: [-0.0172 -1.8920 2.9005 4.0253 3.6242 -7.6514]
vars: [1 8]
c: [0.0210 -1.0050 1.9881 1.7489 1.5632 -3.3096]
vars: [1 9]
c: [0.0011 1.6636 -0.6595 -18.9362 -18.6205 37.5572]
vars: [1 10]
c: [0.1253 -1.2030 2.1164 -2.2638 -2.5810 4.8585]
vars: [1 11]
c: [-0.0079 -2.3364 3.3381 5.5009 5.1177 -10.6199]
vars: [1 12]
c: [0.0574 -0.8055 1.7687 -1.4104 -1.7215 3.1378]
vars: [1 13]
c: [-0.0400 -1.2157 2.2493 4.0642 3.4002 -7.4713]
vars: [1 14]
c: [-0.0096 -1.5630 2.5646 3.3673 3.0905 -6.4584]
vars: [1 15]
c: [0.0090 -1.0731 2.0639 1.9027 1.6389 -3.5406]
vars: [2 3]
c: [0.2105 7.6343 -6.8045 -19.5441 -17.6565 37.2308]
vars: [2 4]
c: [0.0847 -4.5662 5.5098 -22.6115 -23.4799 46.1026]
vars: [2 5]
c: [0.0167 3.0885 -2.1014 -1.5914 -1.0257 2.6198]
vars: [2 6]
c: [0.1116 1.2257 -0.3092 8.4938 8.2263 -16.7120]
vars: [2 7]
c: [-0.0183 -2.0131 3.0196 3.3492 2.5839 -5.9338]
vars: [2 8]
c: [0.0021 -0.6435 1.6423 0.1476 -0.2414 0.0940]
vars: [2 9]
c: [-0.0359 0.4243 0.6091 -6.9645 -6.9224 13.8829]
vars: [2 10]
c: [0.0730 -0.8175 1.7722 -2.1152 -2.4807 4.6028]
vars: [2 11]
c: [-0.0266 -1.8335 2.8472 3.6584 3.0616 -6.7219]
vars: [2 12]
c: [0.0157 -0.2623 1.2557 -1.2872 -1.4452 2.7341]
vars: [2 13]
c: [0.0043 0.3189 0.6770 -0.0099 -0.1354 0.1461]
vars: [2 14]
c: [-0.0239 -1.3359 2.3482 2.2288 1.7019 -3.9324]
vars: [2 15]
c: [-0.0174 -1.2417 2.2511 1.8100 1.2620 -3.0735]
vars: [3 4]
c: [0.1700 -1.6825 2.5518 -8.0689 -8.6765 16.7681]
vars: [3 5]

```

```

c: [-0.0033 2.6919 -1.6879 -0.0072 0.3472 -0.3410]
vars: [3 6]
c: [0.0451 3.4862 -2.5220 16.4591 16.5479 -33.0097]
vars: [3 7]
c: [0.0416 -1.9896 2.9494 3.3804 2.6533 -6.0270]
vars: [3 8]
c: [0.0553 -1.1361 2.0922 0.9994 0.4529 -1.4452]
vars: [3 9]
c: [0.0641 0.0703 0.8826 -3.4584 -3.6624 7.1289]
vars: [3 10]
c: [0.1320 -0.9173 1.8249 -2.0334 -2.4453 4.4935]
vars: [3 11]
c: [0.0486 -2.3562 3.3113 3.8984 3.0576 -6.9485]
vars: [3 12]
c: [0.0286 -0.4945 1.4774 -0.7599 -1.0648 1.8276]
vars: [3 13]
c: [0.1148 0.4902 0.4201 9.7970 9.1432 -18.9306]
vars: [3 14]
c: [0.0448 -1.9467 2.9050 2.6581 1.8512 -4.5022]
vars: [3 15]
c: [0.0506 -1.5357 2.4919 2.0136 1.3596 -3.3660]
vars: [4 5]
c: [0.0348 1.9563 -0.9829 0.7232 0.9472 -1.6662]
vars: [4 6]
c: [-0.0468 -1.5269 2.5675 8.5732 7.7679 -16.3510]
vars: [4 7]
c: [-0.1649 -4.8343 5.9466 10.3053 9.3197 -19.6439]
vars: [4 8]
c: [-0.0788 -3.1872 4.2413 8.4509 7.9041 -16.3643]
vars: [4 9]
c: [0.0199 3.0346 -2.0434 -12.6330 -11.9094 24.5446]
vars: [4 10]
c: [0.0493 -0.6766 1.6471 -2.0114 -2.2397 4.2553]
vars: [4 11]
c: [-0.1822 -4.5087 5.6392 9.9321 9.0858 -19.0404]
vars: [4 12]
c: [0.0332 -0.4762 1.4549 -1.2348 -1.4456 2.6842]
vars: [4 13]
c: [-0.0433 -0.9756 2.0081 -0.6105 -1.0125 1.6183]
vars: [4 14]
c: [-0.1645 -4.2861 5.4008 8.0895 7.2691 -15.3781]
vars: [4 15]
c: [-0.1100 -2.8541 3.9331 5.5946 5.0014 -10.6101]
vars: [5 6]
c: [0.1177 16.3021 -15.4837 79.7443 82.6010 -162.3134]
vars: [5 7]
c: [0.0279 -2.8749 3.8483 6.5084 5.9916 -12.4982]
vars: [5 8]
c: [0.0628 -2.5856 3.5380 6.2465 5.8430 -12.0844]
vars: [5 9]
c: [0.0053 0.5712 0.4246 -4.7932 -4.6620 9.4567]
vars: [5 10]
c: [0.0805 -0.9436 1.8913 -2.2439 -2.5138 4.7658]
vars: [5 11]
c: [0.0360 -3.2852 4.2579 6.9445 6.3953 -13.3393]
vars: [5 12]
c: [0.0170 -0.6304 1.6243 -0.9695 -1.2548 2.2249]
vars: [5 13]
c: [-0.0683 3.1757 -2.1257 20.0214 20.4548 -40.4895]
vars: [5 14]
c: [0.0188 -2.8190 3.8005 5.4704 4.9601 -10.4301]
vars: [5 15]

```

```

c: [0.0363 -2.1512 3.1230 4.0353 3.6140 -7.6475]
vars: [6 7]
c: [0.0049 -3.2734 4.2685 8.8721 8.4149 -17.2912]
vars: [6 8]
c: [0.0609 -1.8114 2.7668 8.6548 8.5653 -17.2179]
vars: [6 9]
c: [-0.0354 1.3789 -0.3519 0.0870 0.4008 -0.4923]
vars: [6 10]
c: [0.0137 -0.2488 1.2422 -1.5308 -1.6505 3.1826]
vars: [6 11]
c: [0.0111 -1.7787 2.7716 5.7882 5.5783 -11.3697]
vars: [6 12]
c: [0.0291 -0.3726 1.3549 -1.5480 -1.7180 3.2689]
vars: [6 13]
c: [0.0588 2.1707 -1.2244 -10.2203 -10.0571 20.2896]
vars: [6 14]
c: [-0.0030 -3.4789 4.4822 7.5478 7.0008 -14.5545]
vars: [6 15]
c: [0.0239 -2.3743 3.3598 5.3362 4.9487 -10.2875]
vars: [7 8]
c: [0.0520 14.8925 -13.9408 90.3349 94.0444 -184.3737]
vars: [7 9]
c: [-0.0240 1.6663 -0.6492 0.6711 0.8947 -1.5686]
vars: [7 10]
c: [0.0445 -0.0971 1.0646 -3.0777 -3.1856 6.2693]
vars: [7 11]
c: [0.0949 8.8766 -7.9487 6.9699 9.0088 -15.9677]
vars: [7 12]
c: [0.0163 0.3856 0.6004 -1.0112 -1.0279 2.0424]
vars: [7 13]
c: [0.0261 1.7623 -0.7818 0.9215 1.0995 -2.0187]
vars: [7 14]
c: [-0.0273 1.8328 -0.8112 12.6294 12.9639 -25.5977]
vars: [7 15]
c: [-0.0374 0.8502 0.1801 10.4064 10.4430 -20.8554]
vars: [8 9]
c: [0.0068 1.3364 -0.3412 -0.1948 -0.0631 0.2587]
vars: [8 10]
c: [0.1627 -2.6052 3.4877 -7.2638 -7.9275 15.2111]
vars: [8 11]
c: [-0.0717 -4.5631 5.6173 51.4183 50.5053 -101.9339]
vars: [8 12]
c: [0.0263 -0.3467 1.3266 -1.5140 -1.6769 3.1954]
vars: [8 13]
c: [0.0202 1.2395 -0.2548 0.7391 0.7834 -1.5206]
vars: [8 14]
c: [-0.0640 -11.6542 12.6906 59.9404 57.3186 -117.2660]
vars: [8 15]
c: [-0.0405 -5.7600 6.7891 34.1312 32.8594 -66.9974]
vars: [9 10]
c: [0.0757 -1.2785 2.2314 -1.7305 -2.1472 3.8847]
vars: [9 11]
c: [-0.0120 -0.5581 1.5653 1.2200 1.0463 -2.2676]
vars: [9 12]
c: [0.0520 -0.9791 1.9502 -1.7240 -2.0758 3.8045]
vars: [9 13]
c: [-0.0349 -0.7453 1.7718 -2.2179 -2.5657 4.7805]
vars: [9 14]
c: [-0.0303 -0.5855 1.6055 1.3157 1.1373 -2.4563]
vars: [9 15]
c: [-0.0204 -0.5061 1.5202 0.8231 0.6204 -1.4459]
vars: [10 11]

```



```

c: [0.1934 3.0020 -2.1445 -6.2776 -5.7650 12.0672]
vars: [10 12]
c: [0.0177 -0.4790 1.4639 -2.3533 -2.5303 4.8873]
vars: [10 13]
c: [0.1018 1.9374 -1.0110 -2.3287 -2.0083 4.3499]
vars: [10 14]
c: [0.0626 0.6834 0.2686 -1.9248 -1.8656 3.7988]
vars: [10 15]
c: [0.0477 0.1412 0.8210 -1.5191 -1.6167 3.1428]
vars: [11 12]
c: [0.0423 -0.4605 1.4291 -2.5008 -2.6239 5.1318]
vars: [11 13]
c: [0.0267 1.2246 -0.2453 0.7893 0.8256 -1.6122]
vars: [11 14]
c: [0.1154 -6.4940 7.4100 -7.9244 -10.4601 18.3978]
vars: [11 15]
c: [0.3484 -13.6123 14.3705 -66.8834 -71.1388 138.0604]
vars: [12 13]
c: [0.0796 0.7810 0.1659 -2.1036 -2.3884 4.5002]
vars: [12 14]
c: [0.0295 0.5914 0.3848 -0.6262 -0.5854 1.2161]
vars: [12 15]
c: [0.0309 0.3774 0.5969 -0.6215 -0.6577 1.2843]
vars: [13 14]
c: [0.0242 -0.7630 1.7442 1.6829 1.5740 -3.2551]
vars: [13 15]
c: [0.0421 -0.6048 1.5724 0.8266 0.6781 -1.5002]
vars: [14 15]
c: [0.0407 -10.4992 11.4659 -75.6759 -79.2317 154.9145]

```

Layer 18: Neurons = 15, Min Error = 0.2931

```

vars: []
c: [2.1494 -0.9850 -3.8397 5.8048 -1.7045 2.8797]
vars: [1 2]
c: [0.1632 10.3413 -9.4444 -36.1562 -33.1330 69.3062]
vars: [1 3]
c: [-0.1598 -6.7342 7.8526 -6.9584 -8.7589 15.7037]
vars: [1 4]
c: [-0.1562 2.7224 -1.6018 1.6915 2.1862 -3.8979]
vars: [1 5]
c: [-0.0293 1.8101 -0.7882 -1.9830 -1.6556 3.6361]
vars: [1 6]
c: [-0.1524 6.4123 -5.2986 -4.8558 -3.4563 8.2958]
vars: [1 7]
c: [-0.0946 3.4371 -2.3642 -5.0127 -4.1355 9.1372]
vars: [1 8]
c: [-0.0625 0.2384 0.8100 -0.0238 -0.0465 0.0624]
vars: [1 9]
c: [-0.2545 -7.4229 8.6212 -9.4375 -11.5695 20.9802]
vars: [1 10]
c: [-0.1197 -2.2006 3.2978 -2.4164 -3.1246 5.5264]
vars: [1 11]
c: [-0.0327 1.3577 -0.3275 -3.3685 -3.0424 6.4061]
vars: [1 12]
c: [-0.2827 3.7105 -2.4866 6.4985 7.0774 -13.6167]
vars: [1 13]
c: [-0.0607 -0.0536 1.1042 -0.6183 -0.7803 1.3902]
vars: [1 14]
c: [-0.0535 1.5778 -0.5320 -2.5830 -2.1925 4.7681]
vars: [1 15]
c: [-0.0756 2.1161 -1.0591 -1.7370 -1.3371 3.0659]
vars: [2 3]

```

```

c: [-0.1740 -4.3107 5.4434 -3.9289 -5.1545 9.0656]
vars: [2 4]
c: [-0.2221 4.4627 -3.3066 12.2431 12.9453 -25.2151]
vars: [2 5]
c: [-0.0723 1.5972 -0.5402 1.5517 1.7790 -3.3406]
vars: [2 6]
c: [-0.0960 4.6875 -3.6141 -3.3357 -2.2789 5.6035]
vars: [2 7]
c: [-0.1535 4.1741 -3.0669 13.6660 14.6209 -28.3044]
vars: [2 8]
c: [-0.0590 -0.0795 1.1252 0.1339 -0.0598 -0.0815]
vars: [2 9]
c: [-0.1678 -5.2748 6.4114 -8.1305 -9.6907 17.8040]
vars: [2 10]
c: [-0.0224 -3.5686 4.5974 -5.4487 -6.4329 11.8795]
vars: [2 11]
c: [-0.0206 0.0650 0.9543 -1.0236 -1.1592 2.1795]
vars: [2 12]
c: [-0.2249 4.0006 -2.8400 11.3076 11.9066 -23.2450]
vars: [2 13]
c: [-0.0476 -0.6656 1.7067 -0.2871 -0.7621 1.0422]
vars: [2 14]
c: [-0.0464 -0.1906 1.2275 -0.0338 -0.3272 0.3549]
vars: [2 15]
c: [0.0242 0.5935 0.3857 -3.1050 -3.0955 6.2054]
vars: [3 4]
c: [-0.3499 7.8170 -6.5507 -4.3760 -2.6190 6.9558]
vars: [3 5]
c: [-0.2107 6.5476 -5.3908 -7.8651 -6.2620 14.1079]
vars: [3 6]
c: [-0.2569 5.4141 -4.2190 -2.7741 -1.6568 4.4027]
vars: [3 7]
c: [-0.3670 10.5542 -9.2774 -7.2414 -4.9423 12.1441]
vars: [3 8]
c: [-0.1562 1.9996 -0.8757 -1.8706 -1.4696 3.3214]
vars: [3 9]
c: [-0.2213 -2.7632 3.9391 2.9486 2.1489 -5.1278]
vars: [3 10]
c: [-0.2456 -0.4538 1.6363 3.3800 3.0688 -6.4784]
vars: [3 11]
c: [-0.1867 3.6953 -2.5485 -4.9339 -4.1471 9.0603]
vars: [3 12]
c: [-0.4857 8.3443 -6.9670 -1.2373 0.5214 0.6556]
vars: [3 13]
c: [-0.1785 1.2992 -0.1586 -1.0006 -0.7773 1.7556]
vars: [3 14]
c: [-0.2275 5.1405 -3.9583 -7.1696 -6.0537 13.1977]
vars: [3 15]
c: [-0.5782 11.0838 -9.6374 -3.9312 -1.5494 5.4115]
vars: [4 5]
c: [0.0899 2.9088 -1.9551 -17.6056 -17.3902 35.0035]
vars: [4 6]
c: [0.0111 1.9542 -0.9597 -3.4970 -3.1355 6.6344]
vars: [4 7]
c: [-0.0540 -3.7574 4.8016 12.3917 11.4601 -23.8611]
vars: [4 8]
c: [-0.1015 -1.1556 2.2286 2.3465 1.8893 -4.2478]
vars: [4 9]
c: [-0.3249 -7.4454 8.6868 -3.9664 -6.0015 9.9349]
vars: [4 10]
c: [-0.2532 -2.2625 3.4594 2.0100 1.2890 -3.3320]
vars: [4 11]

```

```

c: [-0.0881 -1.2428 2.3068 3.8976 3.4491 -7.3573]
vars: [4 12]
c: [0.0305 -1.9090 2.8820 -4.8303 -5.4383 10.2749]
vars: [4 13]
c: [-0.0667 -0.9253 1.9766 0.7306 0.3242 -1.0632]
vars: [4 14]
c: [-0.2144 -2.6126 3.7709 8.3656 7.7222 -16.1147]
vars: [4 15]
c: [-0.1317 -0.4057 1.5251 15.7811 16.0143 -31.8231]
vars: [5 6]
c: [0.0096 1.5707 -0.5791 -1.6282 -1.2985 2.9289]
vars: [5 7]
c: [0.0687 -1.1652 2.1146 -8.4124 -8.6890 17.1118]
vars: [5 8]
c: [-0.0562 -0.0988 1.1417 0.2388 0.0966 -0.3423]
vars: [5 9]
c: [-0.0176 -5.3874 6.3954 -7.5055 -9.1824 16.6939]
vars: [5 10]
c: [-0.0948 -3.3691 4.4503 -3.5846 -4.6156 8.1888]
vars: [5 11]
c: [-0.0036 -0.1280 1.1331 -0.9846 -1.1085 2.0923]
vars: [5 12]
c: [-0.0477 1.1012 -0.0669 1.8320 1.9247 -3.7627]
vars: [5 13]
c: [-0.0412 -0.4175 1.4522 -0.3474 -0.5987 0.9405]
vars: [5 14]
c: [-0.0537 -0.1401 1.1810 0.5684 0.4277 -1.0028]
vars: [5 15]
c: [-0.0247 0.3694 0.6488 -1.6805 -1.7099 3.3883]
vars: [6 7]
c: [0.0066 -3.8333 4.8306 -13.8338 -14.5382 28.3760]
vars: [6 8]
c: [-0.0197 -0.5880 1.6054 -0.1504 -0.5313 0.6789]
vars: [6 9]
c: [-0.2598 -4.0159 5.2144 -2.7201 -3.8548 6.5470]
vars: [6 10]
c: [-0.1575 -2.5137 3.6378 -1.7485 -2.4489 4.1795]
vars: [6 11]
c: [0.0195 -0.8601 1.8515 -2.6067 -2.9748 5.5832]
vars: [6 12]
c: [0.0398 0.0323 0.9356 -0.0484 -0.2164 0.2704]
vars: [6 13]
c: [-0.0134 -0.9267 1.9432 -0.8273 -1.2969 2.1215]
vars: [6 14]
c: [-0.0218 -0.8164 1.8361 -0.7803 -1.2544 2.0316]
vars: [6 15]
c: [0.3365 -10.0596 10.8104 -22.5970 -25.2516 47.8928]
vars: [7 8]
c: [-0.0804 -1.3005 2.3574 2.1243 1.5486 -3.6819]
vars: [7 9]
c: [-0.3737 -8.7547 10.0386 -3.1542 -5.4981 8.6104]
vars: [7 10]
c: [-0.1793 -2.6883 3.8293 -0.0673 -0.8906 0.9354]
vars: [7 11]
c: [-0.0548 -1.3868 2.4284 1.4091 0.6800 -2.0959]
vars: [7 12]
c: [0.0246 1.2449 -0.2615 5.0973 5.1037 -10.2004]
vars: [7 13]
c: [-0.0707 -1.2063 2.2616 0.5994 -0.0510 -0.5574]
vars: [7 14]
c: [-0.1356 -3.2500 4.3460 6.7321 5.6665 -12.4138]
vars: [7 15]

```

```

      c: [0.0624 0.3748 0.5723 -3.1366 -3.2222 6.3690]
vars: [8 9]
      c: [-0.1841 -2.5954 3.7474 -3.3265 -4.2336 7.5369]
vars: [8 10]
      c: [-0.1992 -3.1181 4.2700 -3.7316 -4.6349 8.3447]
vars: [8 11]
      c: [-0.0704 10.6730 -9.6336 44.6127 47.4924 -92.1124]
vars: [8 12]
      c: [-0.2059 3.3961 -2.2532 6.0000 6.5573 -12.5811]
vars: [8 13]
      c: [0.0364 -8.1835 9.1547 -35.3952 -37.6167 73.0195]
vars: [8 14]
      c: [0.0369 3.2488 -2.2757 -1.5063 -0.8551 2.3656]
vars: [8 15]
      c: [-0.0628 1.2803 -0.2310 -0.0992 0.1000 -0.0089]
vars: [9 10]
      c: [-0.2455 -1.0170 2.1993 6.4643 6.1297 -12.6259]
vars: [9 11]
      c: [-0.2520 7.6316 -6.4301 -11.6350 -9.6884 21.2965]
vars: [9 12]
      c: [-0.6094 9.8282 -8.3539 -0.9666 1.0981 -0.2078]
vars: [9 13]
      c: [-0.2095 1.8995 -0.7351 -1.2927 -0.8653 2.1318]
vars: [9 14]
      c: [-0.3095 9.2224 -7.9670 -13.7495 -11.4540 25.1676]
vars: [9 15]
      c: [-0.4046 8.6785 -7.3709 -5.0226 -3.1034 8.0807]
vars: [10 11]
      c: [-0.2760 7.4687 -6.2557 -4.1567 -2.5514 6.6758]
vars: [10 12]
      c: [-0.4676 3.8734 -2.5114 6.5442 7.2508 -13.8595]
vars: [10 13]
      c: [-0.2128 3.4696 -2.3034 -5.1213 -4.4281 9.5245]
vars: [10 14]
      c: [-0.2291 5.9143 -4.7369 -5.8932 -4.6487 10.5159]
vars: [10 15]
      c: [-0.2145 3.5001 -2.3313 -1.7544 -1.0504 2.7787]
vars: [11 12]
      c: [-0.1167 3.4218 -2.3421 12.5093 12.8667 -25.3925]
vars: [11 13]
      c: [0.0938 -5.6126 6.5469 -7.2539 -9.0056 16.2699]
vars: [11 14]
      c: [-0.1146 -5.7133 6.8016 48.0596 46.7435 -94.8188]
vars: [11 15]
      c: [-0.0072 1.1484 -0.1416 -1.1393 -0.9892 2.1277]
vars: [12 13]
      c: [-0.1397 -0.9427 2.0450 3.1977 2.8970 -6.1123]
vars: [12 14]
      c: [-0.2789 -3.8558 5.0654 13.1890 12.4375 -25.6659]
vars: [12 15]
      c: [-0.0222 0.1549 0.8643 1.1875 1.1881 -2.3798]
vars: [13 14]
      c: [0.2446 13.1825 -12.3425 -28.3474 -24.5326 52.9041]
vars: [13 15]
      c: [-0.0631 1.5299 -0.4780 -0.6560 -0.3742 1.0219]
vars: [14 15]
      c: [-0.0634 1.3783 -0.3285 -0.0956 0.1242 -0.0367]

Layer 19: Neurons = 15, Min Error = 0.28999
vars: []
      c: [0.0049 0.0099 0.0509 0.0148 0.0461 0.0114]
vars: [1 2]

```

```

c: [-0.0442 1.0001 0.0326 -0.2475 -0.1369 0.3797]
vars: [1 3]
c: [7.2542e-04 2.0442 -1.0469 -0.1664 0.1846 -0.0173]
vars: [1 4]
c: [-0.0209 1.8488 -0.8342 -0.2128 0.1001 0.1108]
vars: [1 5]
c: [0.0092 6.5859 -5.5923 -0.1536 1.3596 -1.2050]
vars: [1 6]
c: [-0.0164 4.8325 -3.8186 0.7383 1.7670 -2.5080]
vars: [1 7]
c: [-0.0157 1.9030 -0.8928 -0.1156 0.2088 -0.0946]
vars: [1 8]
c: [-0.0532 0.9530 0.0872 0.0323 0.0827 -0.1214]
vars: [1 9]
c: [-0.0172 1.7575 -0.7452 0.3298 0.5672 -0.8990]
vars: [1 10]
c: [-0.1893 -0.7129 1.8626 -0.9375 -1.1559 2.0712]
vars: [1 11]
c: [-0.0022 2.6075 -1.6055 1.0118 1.3537 -2.3665]
vars: [1 12]
c: [-0.0423 1.1637 -0.1324 -0.1858 -0.0495 0.2308]
vars: [1 13]
c: [-0.0557 1.4364 -0.3950 -0.6044 -0.3430 0.9417]
vars: [1 14]
c: [-0.0446 0.7167 0.3164 -0.2844 -0.2373 0.5170]
vars: [1 15]
c: [-0.0802 0.7027 0.3582 -0.2046 -0.1608 0.3561]
vars: [2 3]
c: [3.4324e-04 0.8474 0.1519 0.1367 0.1612 -0.2977]
vars: [2 4]
c: [-0.0177 1.7526 -0.7407 1.0566 1.3401 -2.3986]
vars: [2 5]
c: [-0.1512 4.4176 -3.2999 4.3682 5.5619 -9.9509]
vars: [2 6]
c: [-0.0209 2.0936 -1.0753 -0.2585 0.1972 0.0580]
vars: [2 7]
c: [0.0192 0.6755 0.3112 -1.0766 -0.9329 2.0118]
vars: [2 8]
c: [-0.0019 7.9508 -6.9468 27.1745 28.8072 -55.9835]
vars: [2 9]
c: [0.0289 0.6828 0.2947 -0.7423 -0.6829 1.4291]
vars: [2 10]
c: [-0.0955 -4.7638 5.8936 -6.2511 -7.1781 13.4092]
vars: [2 11]
c: [-0.0253 1.7384 -0.7191 0.5704 0.8840 -1.4578]
vars: [2 12]
c: [0.0350 -2.2154 3.1941 -14.5087 -15.0877 29.6002]
vars: [2 13]
c: [0.0491 -4.0005 4.9734 -12.9843 -13.9671 26.9558]
vars: [2 14]
c: [0.1125 -6.8993 7.8394 -18.5949 -20.4255 39.0287]
vars: [2 15]
c: [0.0097 0.3772 0.6194 -0.9552 -0.9023 1.8584]
vars: [3 4]
c: [0.1883 6.9195 -6.0583 -22.4466 -20.4406 42.9127]
vars: [3 5]
c: [-0.0862 3.4760 -2.4065 -0.5847 0.2440 0.3291]
vars: [3 6]
c: [-0.0672 3.0318 -1.9779 -0.8278 -0.1734 0.9924]
vars: [3 7]
c: [0.2043 1.5539 -0.7239 -13.5489 -12.8855 26.4665]
vars: [3 8]

```

```

c: [0.0024 0.2518 0.7461 0.0141 -0.0220 0.0083]
vars: [3 9]
c: [0.0115 1.5490 -0.5591 -0.7939 -0.5146 1.3105]
vars: [3 10]
c: [-0.0496 -5.0874 6.1723 -6.2932 -7.3692 13.6493]
vars: [3 11]
c: [-0.0197 1.7031 -0.6873 -2.7997 -2.4809 5.2789]
vars: [3 12]
c: [0.0171 1.8670 -0.8761 -3.3955 -2.9714 6.3685]
vars: [3 13]
c: [-0.0161 1.5287 -0.5103 -2.6135 -2.2731 4.8834]
vars: [3 14]
c: [0.0083 0.2944 0.7017 -1.0757 -1.1069 2.1833]
vars: [3 15]
c: [0.0595 0.7199 0.2367 -1.9059 -1.7396 3.6538]
vars: [4 5]
c: [-0.1750 4.5690 -3.4373 11.6535 13.1717 -24.8489]
vars: [4 6]
c: [-0.0277 1.2528 -0.2307 0.4558 0.7655 -1.2253]
vars: [4 7]
c: [0.0589 -0.2854 1.2468 -4.0212 -3.8923 7.9194]
vars: [4 8]
c: [-0.0118 -0.5003 1.5087 0.7354 0.4789 -1.2157]
vars: [4 9]
c: [0.1085 -1.7322 2.6491 -5.7468 -6.3835 12.1458]
vars: [4 10]
c: [0.0810 -5.9528 6.9469 -5.0389 -6.3524 11.3913]
vars: [4 11]
c: [0.2003 -3.1936 4.0340 -18.9775 -20.0312 39.0397]
vars: [4 12]
c: [-0.0232 -0.6999 1.7163 1.4465 1.1551 -2.6042]
vars: [4 13]
c: [-0.0020 -0.1211 1.1233 -0.1521 -0.4015 0.5532]
vars: [4 14]
c: [-0.0042 -0.3855 1.3906 -0.2425 -0.5202 0.7617]
vars: [4 15]
c: [0.0795 -0.5156 1.4596 -3.6566 -3.7317 7.3980]
vars: [5 6]
c: [-0.0787 -1.0819 2.1424 -13.0155 -13.8725 26.8799]
vars: [5 7]
c: [-0.0875 -1.3252 2.3882 8.0997 7.4291 -15.5403]
vars: [5 8]
c: [-0.1752 -3.6096 4.7470 6.4530 5.1504 -11.6276]
vars: [5 9]
c: [-0.0886 -2.5708 3.6380 8.2516 7.5887 -15.8529]
vars: [5 10]
c: [-0.0643 -3.9186 4.9946 -4.0629 -4.9344 8.9877]
vars: [5 11]
c: [-0.0364 -1.3511 2.3805 -2.6588 -3.4925 6.1465]
vars: [5 12]
c: [-0.2270 -4.0851 5.2610 9.0209 7.5725 -16.6241]
vars: [5 13]
c: [-0.0638 -1.6188 2.6692 2.0508 1.2950 -3.3547]
vars: [5 14]
c: [-0.1157 -1.8458 2.9381 2.1676 1.3752 -3.5589]
vars: [5 15]
c: [-0.1269 -0.9909 2.0753 4.0232 3.5827 -7.6196]
vars: [6 7]
c: [-0.0178 -0.3190 1.3329 0.4166 0.1559 -0.5748]
vars: [6 8]
c: [-0.0451 -1.3168 2.3544 0.6447 0.0815 -0.7329]
vars: [6 9]

```

```

c: [-0.0559 -1.1595 2.2041 1.3239 0.8234 -2.1551]
vars: [6 10]
c: [-0.1134 -2.6478 3.7580 -3.1731 -3.6803 6.8385]
vars: [6 11]
c: [-0.0510 -0.5989 1.6385 -3.6418 -3.9235 7.5597]
vars: [6 12]
c: [-0.0441 -0.9084 1.9442 0.5685 0.1154 -0.6902]
vars: [6 13]
c: [-0.0189 -0.2309 1.2465 -0.5414 -0.7637 1.3028]
vars: [6 14]
c: [-0.0107 -0.7663 1.7768 -0.3040 -0.6521 0.9543]
vars: [6 15]
c: [-0.0173 -0.0473 1.0603 -0.0260 -0.1705 0.1945]
vars: [7 8]
c: [0.0482 0.6956 0.2689 -1.8162 -1.8980 3.7203]
vars: [7 9]
c: [0.0606 -0.2355 1.1849 -1.9092 -2.1294 4.0479]
vars: [7 10]
c: [-0.1279 -4.1465 5.2787 -5.3929 -6.3267 11.6998]
vars: [7 11]
c: [0.0605 0.8706 0.0834 -2.0943 -2.0715 4.1739]
vars: [7 12]
c: [0.0675 0.8867 0.0682 -3.1764 -3.2986 6.4823]
vars: [7 13]
c: [0.0240 0.5575 0.4297 -1.8775 -2.0240 3.9034]
vars: [7 14]
c: [0.0226 0.0998 0.8868 -1.4633 -1.6946 3.1602]
vars: [7 15]
c: [-0.0377 0.5735 0.4586 -0.6587 -0.5803 1.2337]
vars: [8 9]
c: [0.0228 0.3072 0.6759 -1.5267 -1.5276 3.0574]
vars: [8 10]
c: [-0.0978 -4.4054 5.5283 -6.0722 -6.9347 12.9885]
vars: [8 11]
c: [-0.0469 2.0046 -0.9684 0.8207 1.2543 -2.0812]
vars: [8 12]
c: [0.0934 -1.2474 2.1810 -25.1147 -25.3121 50.4383]
vars: [8 13]
c: [0.0807 -3.0255 3.9690 -14.7504 -15.5329 30.2935]
vars: [8 14]
c: [0.1337 -4.3232 5.2385 -19.4757 -20.4268 39.9159]
vars: [8 15]
c: [0.0040 0.4283 0.5721 -0.8773 -0.8080 1.6855]
vars: [9 10]
c: [-0.0857 -3.7389 4.8355 -4.7682 -5.5173 10.2735]
vars: [9 11]
c: [0.0138 0.4311 0.5612 -2.2718 -2.3685 4.6416]
vars: [9 12]
c: [0.0495 0.9503 0.0132 -2.1167 -1.9644 4.0875]
vars: [9 13]
c: [0.0283 0.6501 0.3307 -1.7067 -1.6733 3.3836]
vars: [9 14]
c: [0.0418 0.1936 0.7766 -1.4098 -1.4740 2.8889]
vars: [9 15]
c: [0.0293 0.4527 0.5274 -1.0547 -1.0169 2.0754]
vars: [10 11]
c: [0.0262 5.1943 -4.1880 -4.5164 -3.5927 8.1102]
vars: [10 12]
c: [-0.0348 5.5798 -4.5133 -6.2745 -5.3698 11.6347]
vars: [10 13]
c: [-0.0571 6.1251 -5.0399 -7.6467 -6.6737 14.3079]
vars: [10 14]

```

```

c: [-0.1535 5.0261 -3.8820 -7.6359 -6.9224 14.5382]
vars: [10 15]
c: [-0.0722 3.8140 -2.7378 -2.5260 -1.7712 4.2857]
vars: [11 12]
c: [-0.1122 -2.3339 3.4201 4.8659 4.1000 -8.9808]
vars: [11 13]
c: [-0.0204 -0.5021 1.5174 0.9790 0.7117 -1.6934]
vars: [11 14]
c: [-0.0244 -0.6504 1.6700 0.4953 0.1848 -0.6835]
vars: [11 15]
c: [-0.0666 -0.4575 1.5072 1.7155 1.5316 -3.2558]
vars: [12 13]
c: [0.0441 0.6908 0.2739 -6.9419 -6.9740 13.9223]
vars: [12 14]
c: [0.2055 -6.7912 7.6499 -20.7104 -22.4421 43.1747]
vars: [12 15]
c: [0.0152 0.2390 0.7520 -0.9795 -0.9349 1.9160]
vars: [13 14]
c: [0.1943 -8.1931 9.0494 -25.7889 -27.6980 53.5115]
vars: [13 15]
c: [-0.0247 0.3642 0.6557 -0.3632 -0.3235 0.6836]
vars: [14 15]
c: [-0.0503 0.7148 0.3252 -0.3476 -0.2364 0.5776]

```

Layer 20: Neurons = 1, Min Error = 0.29062

Berikut merupakan estimasi koefisien pada setiap neuron dan setiap lapisan pada model ke-3

```

vars: []
c: [1.1023 -0.1714 0.2565 -0.9239 -0.5194 0.0508]
vars: [1 2]
c: [2.1455 -0.5143 -0.0689 0.2727 0.1053 -0.1607]
vars: [1 3]
c: [0.4593 -0.2753 1.1271 0.2029 -0.1513 -0.0698]
vars: [1 4]
c: [1.6002 -0.2410 0.1095 0.2027 0.0294 -0.0901]
vars: [1 5]
c: [1.0785 -0.2864 0.5665 0.1767 -0.0652 -0.0258]
vars: [1 6]
c: [1.5705 0.0793 -0.1267 0.1487 0.0470 -0.0765]
vars: [1 7]
c: [0.6915 -0.0685 0.7065 0.1839 -0.0648 -0.1124]
vars: [2 3]
c: [1.4105 -0.8316 1.0467 0.3003 -0.0935 -0.1484]
vars: [2 4]
c: [1.4205 -0.5388 0.7360 0.2265 -0.0652 -0.0997]
vars: [2 5]
c: [1.3478 -0.6638 0.8543 0.2588 -0.0563 -0.1373]
vars: [2 6]
c: [1.5737 -0.1380 0.1988 0.1823 0.0360 -0.1445]
vars: [2 7]
c: [1.1717 -0.0868 0.5175 0.1744 -0.0262 -0.1396]
vars: [3 4]
c: [1.0524 0.3145 0.3284 0.0761 0.0350 -0.1449]
vars: [3 5]
c: [0.2779 0.2734 0.9657 0.0744 -0.0673 -0.1345]
vars: [3 6]
c: [0.3900 0.6852 0.4683 0.0680 0.0425 -0.2433]
vars: [3 7]

```



```

c: [-0.5853 1.2146 0.7703 -0.0272 -0.0188 -0.2268]
vars: [4 5]
c: [1.4855 -0.4485 0.7707 0.1947 0.0082 -0.1935]
vars: [4 6]
c: [0.8440 0.5701 0.3106 0.0805 0.0811 -0.2596]
vars: [4 7]
c: [0.1771 0.3498 1.0521 0.1439 -0.0201 -0.3172]
vars: [5 6]
c: [0.3720 1.0343 0.3364 -0.0719 -0.0437 -0.0568]
vars: [5 7]
c: [-0.1858 1.1249 0.6796 -0.0459 -0.0375 -0.1667]
vars: [6 7]
c: [1.4754 -0.0033 0.5311 0.1142 0.0142 -0.1825]

```

Layer 1: Neurons = 6, Min Error = 0.39362

```

vars: []
c: [0.8028 -0.7609 0.2515 -0.8629 1.4639 0.6422]
vars: [1 2]
c: [-0.1492 0.2922 0.8210 0.2773 0.1966 -0.4929]
vars: [1 3]
c: [-0.3851 -0.4677 1.7677 0.0425 -0.2864 0.1947]
vars: [1 4]
c: [-0.0483 1.7244 -0.7161 3.7698 4.0721 -7.8516]
vars: [1 5]
c: [-0.3239 0.4278 0.8304 0.9012 0.8528 -1.8034]
vars: [1 6]
c: [-0.3889 0.1741 1.1306 0.8585 0.7853 -1.7018]
vars: [2 3]
c: [-0.4049 -0.3986 1.7130 0.0711 -0.2461 0.1232]
vars: [2 4]
c: [0.0423 1.3969 -0.4409 4.3144 4.6229 -8.9409]
vars: [2 5]
c: [-0.8058 -0.4212 2.0457 3.2444 2.9654 -6.3275]
vars: [2 6]
c: [-0.8124 0.7568 0.8983 1.8357 1.9512 -3.9138]
vars: [3 4]
c: [-0.0537 1.3115 -0.3143 1.6190 1.9540 -3.5729]
vars: [3 5]
c: [-0.7433 -0.6920 2.2457 2.5801 2.3488 -5.0341]
vars: [3 6]
c: [-0.6649 1.3391 0.1690 2.0761 2.3893 -4.5664]
vars: [4 5]
c: [0.7991 -2.3164 2.7259 9.6114 8.9294 -18.4981]
vars: [4 6]
c: [-0.2322 -0.2932 1.4884 1.6809 1.5697 -3.2946]
vars: [5 6]
c: [-0.4134 0.0177 1.2781 3.7744 3.4117 -7.2443]

```

Layer 2: Neurons = 3, Min Error = 0.41164

```

vars: []
c: [0.7280 0.1792 -0.3897 -0.0338 -0.2017 -0.2852]
vars: [1 2]
c: [-0.2342 1.6043 -0.4405 -0.7230 -0.5123 1.2189]
vars: [1 3]
c: [-0.4997 0.6466 0.7281 -0.7210 -0.7771 1.4477]
vars: [2 3]
c: [-0.2597 -1.3301 2.5466 -0.0458 -0.7335 0.7422]

```

Layer 3: Neurons = 1, Min Error = 0.41848



## LAMPIRAN G: Uji MDA

**Tabel G 1 Hasil Output dan Uji MDA terhadap Hasil peramalan  
Model ke-1**

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0				
0				
0				
0				
49700	15,05770077	1		
24794	32306,42994	0	1	0
10600	9834,8787	0	0	1
0	1279,385878	0	0	1
15800	13849,55146	1	1	1
41275	5464,639289	1	0	0
20450	21949,67787	0	1	0
34600	5440,907568	1	0	0
0	7012,884942	0	1	0
43243	21613,8495	1	1	1
36900	19288,25042	0	0	1
12021	13118,40255	0	0	1
8500	1755,729178	0	0	1
0	823,0654397	0	0	1
0	235,2162565	0	0	1
0	19,4715714	0	0	1
0	53,16389146	0	1	0
13500	15,05770077	1	0	0
0	3515,377887	0	1	0
0	377,3697668	0	0	1

## H-2

[illegible]

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
56100	15,05770077	1	0	0
7450	27224,73816	0	1	0
21000	60166,39283	1	1	1
6700	8515,612717	0	0	1
13489	1419,653918	1	0	0
37650	2070,53373	1	1	1
37000	24279,87388	0	1	0
34100	11513,15338	0	0	1
5000	6669,715904	0	0	1
14400	598,0337343	1	0	0
7200	1642,567596	0	1	0
7175	1719,015764	0	1	0
0	1004,387539	0	0	1
0	170,7013662	0	0	1
0	29,98970643	0	0	1
0	70,6782595	0	1	0
0	15,05770077	0	0	1

## H-4

[illegible]

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	15,05770077	0	0	1
0	15,05770077	0	0	1
12945	15,05770077	1	0	0
41610	3177,927446	1	1	1
26099	25685,19859	0	1	0
58115	8641,763652	1	0	0
97544	15058,38695	1	1	1
2997	31297,64657	0	1	0
0	1481,053612	0	0	1
0	37,38609064	0	0	1
0	60,43893658	0	1	0
52450	697,3386458	1	1	1
92089	30058,11289	1	1	1
60738	63440,0812	0	1	0
101599	119366,8523	1	1	1
34934	51583,20757	0	0	1
44502	4990,573902	1	0	0
82830	5879,490824	1	1	1
45010	25828,10598	0	1	0
100	8427,574002	0	0	1
0	134,4445294	0	0	1
0	147,222552	0	1	0
0	43,73601306	0	0	1
0	11,83670366	0	0	1
0	15,05770077	0	1	0
6981	15,05770077	1	0	0
51356	1268,903273	1	1	1

## H-6

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
112036	36712,1676	1	1	1
110716	94705,92272	0	1	0
9677	21707,66052	0	0	1
0	858,5033917	0	0	1
0	32,63612835	0	0	1
0	68,90264922	0	1	0
0	35,38520135	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
46248	15,05770077	1	0	0
23387	34821,80415	0	1	0
0	7807,1135	0	0	1
0	-40	0	0	1
0	60,27602424	0	1	0



Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	-40	0	0	1
0	15,05770077	0	1	0
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
0	15,05770077	0	0	1
22654	15,05770077	1	0	0
18269	20049,97936	0	1	0
57288	6516,589023	1	0	0
70093	64264,83268	1	1	1
137047	21553,7566	1	0	0
56919	37159,95934	0	1	0
1750	9230,598621	0	0	1
13541	308,2089624	1	0	0
40	1338,407029	0	1	0
0	171,5117645	0	0	1
0	51,67665296	0	0	1
0	39,52829669	0	0	1
0	10,27867098	0	0	1
0	15,05770077	0	1	0
			All	0,761627
			Training	0,793388
			Testing	0,685185

**Tabel G 2 Hasil Output dan Uji MDA terhadap Hasil peramalan Model ke-2**

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0				
0				
0				
0				
49700				
24794				
10600	19777,84	0		
0	3311,07	0	0	1
15800	107,57	1	0	0
41275	3346,57	1	1	1
20450	34695,96	0	1	0
34600	11530,39	1	0	0
0	795,14	0	0	1
43243	2759,77	1	1	1
36900	17322,92	0	1	0
12021	32003,21	0	1	0
8500	2264,97	0	0	1
0	216,45	0	0	1
0	74,84	0	0	1
0	58,82	0	0	1
0	54,35	0	0	1
13500	104,84	1	1	1
0	375,66	0	1	0
0	55,32	0	0	1
0	86,35	0	1	0



## H-10

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
56100	11,03	1	0	0
7450	57138,74	0	1	0
21000	11218,86	1	0	0
6700	8180,23	0	0	1
13489	11280,23	1	1	1
37650	2948,95	1	0	0
37000	23780,63	0	1	0
34100	12825,47	0	0	1
5000	4887,80	0	0	1
14400	755,40	1	0	0
7200	2461,44	0	1	0
7175	829,78	0	0	1
0	628,52	0	0	1
0	126,79	0	0	1
0	176,07	0	1	0
0	107,55	0	0	1
0	21,06	0	0	1
0	18,11	0	0	1



## H-12

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	11,03	0	0	1
12945	11,03	1	0	0
41610	24610,80	1	1	1
26099	51973,78	0	1	0
58115	33120,62	1	0	0
97544	48964,99	1	1	1
2997	10645,93	0	0	1
0	201,81	0	0	1
0	41,31	0	0	1
0	45,97	0	1	0
52450	27,01	1	0	0
92089	98750,40	1	1	1
60738	65364,15	0	0	1
101599	131010,53	1	1	1
34934	13803,85	0	0	1
44502	14462,81	1	1	1
82830	11730,37	1	0	0
45010	62060,27	0	1	0
100	7316,90	0	0	1
0	66,39	0	0	1
0	36,64	0	0	1
0	34,73	0	0	1
0	14,92	0	0	1
0	7,38	0	0	1
6981	5,52	1	0	0
51356	40553,79	1	1	1
112036	63661,72	1	1	1



Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	76,53	0	0	1
0	81,20	0	1	0
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
0	11,03	0	0	1
22654	11,03	1	0	0
18269	19256,14	0	1	0
57288	-20,99	1	0	0
70093	51615,21	1	1	1
137047	76030,79	1	1	1
56919	31703,16	0	0	1
1750	-40,00	0	0	1
13541	106,87	1	1	1
40	1177,59	0	1	0
0	134,08	0	0	1
0	-40,00	0	0	1
0	32,84	0	1	0
0	-40,00	0	0	1
0	3,64	0	1	0
			All	0,774566
			Training	0,798319
			Testing	0,722222



**Tabel G 3 Hasil Output dan Uji MDA terhadap Hasil peramalan Model ke-3**

Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0				
0				
0				
0				
49700				
24794				
10600				
0	6790,63	0		
15800	2,49	1	0	0
41275	3784,70	1	1	1
20450	30003,14	0	1	0
34600	11960,31	1	0	0
0	1579,51	0	0	1
43243	221,07	1	0	0
36900	14858,04	0	1	0
12021	26084,94	0	1	0
8500	2684,66	0	0	1
0	240,41	0	0	1
0	168,08	0	0	1
0	33,30	0	0	1
0	60,07	0	1	0
13500	102,84	1	1	1
0	222,88	0	1	0
0	17,39	0	0	1
0	12,58	0	0	1



Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
56100	17,39	1	0	0
7450	62002,96	0	1	0
21000	6128,50	1	0	0
6700	13882,15	0	1	0
13489	5220,67	1	0	0
37650	3335,98	1	0	0
37000	32270,59	0	1	0
34100	14256,60	0	0	1
5000	18331,40	0	1	0
14400	1291,64	1	0	0
7200	2965,31	0	1	0
7175	1278,59	0	0	1
0	1413,06	0	1	0
0	52,21	0	0	1
0	215,22	0	1	0
0	113,77	0	0	1
0	119,96	0	1	0
0	24,61	0	0	1



Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	17,39	0	0	1
12945	17,39	1	0	0
41610	11254,20	1	1	1
26099	43647,13	0	1	0
58115	22814,46	1	0	0
97544	32647,92	1	1	1
2997	20587,00	0	0	1
0	508,40	0	0	1
0	41,93	0	0	1
0	32,71	0	0	1
52450	19,57	1	0	0
92089	61974,57	1	1	1
60738	2440,83	0	0	1
101599	48083,26	1	1	1
34934	47248,33	0	0	1
44502	10340,51	1	0	0
82830	9891,75	1	0	0
45010	47343,35	0	1	0
100	9585,09	0	0	1
0	34,29	0	0	1
0	34,71	0	1	0
0	15,76	0	0	1
0	26,39	0	1	0
0	24,32	0	0	1
6981	4,43	1	0	0
51356	5717,15	1	1	1
112036	55901,06	1	1	1



Data Aktual (y)	Peramalan(f)	Naik(y)	Naik(f)	Akurasi
0	45,32	0	0	1
0	71,01	0	1	0
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
0	17,39	0	0	1
22654	17,39	1	0	0
18269	21384,21	0	1	0
57288	16666,69	1	0	0
70093	51234,28	1	1	1
137047	51047,40	1	0	0
56919	40817,18	0	0	1
1750	26321,05	0	0	1
13541	269,64	1	0	0
40	1928,67	0	1	0
0	4,56	0	0	1
0	16,35	0	1	0
0	21,11	0	1	0
0	146,57	0	1	0
0	11,86	0	0	1
			All	0,715116
			Training	0,754237
			Testing	0,629629





*(Halaman ini sengaja dikosongkan)*



## LAMPIRAN H: Uji Coba Model Berdasarkan Nilai RMSE

**Tabel H 1 Uji Coba Penambahan Maksimum Lapisan**

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
3	3	20	0	Infinite	Infinite
4	6	20	0	16673	Infinite
5	10	20	0	15877	Infinite
6	15	20	0	14448	Infinite
7	21	20	0	16091	Infinite
8	28	20	0	18771	Infinite
9	36	20	0	19059	Infinite
10	45	20	0	15545	Infinite
11	55	20	0	14103	Infinite
12	66	20	0	16935	Infinite
3	3	30	0	Infinite	Infinite
4	6	30	0	16173	Infinite
5	10	30	0	15708	Infinite
6	15	30	0	15723	Infinite
7	21	30	0	14047	Infinite
8	28	30	0	17326	Infinite
9	36	30	0	21239	Infinite
10	45	30	0	14885	Infinite
11	55	30	0	12134	Infinite
12	66	30	0	23005	Infinite
3	3	40	0	Infinite	Infinite
4	6	40	0	15715	Infinite
5	10	40	0	15152	Infinite
6	15	40	0	14326	Infinite
7	21	40	0	16841	Infinite
8	28	40	0	21194	Infinite
9	36	40	0	20445	Infinite
10	45	40	0	15575	Infinite

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
11	55	40	0	11719	Infinite
12	66	40	0	14912	Infinite
3	3	50	0	Infinite	Infinite
4	6	50	0	19849	Infinite
5	10	50	0	14997	Infinite
6	15	50	0	14356	Infinite
7	21	50	0	15664	Infinite
8	28	50	0	56936	Infinite
9	36	50	0	22506	Infinite
10	45	50	0	14073	Infinite
11	55	50	0	11236	Infinite
12	66	50	0	19838	Infinite

**Tabel H 2 Uji Coba Penambahan *Selection Pressure* 0,1-0,5**

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
3	3	10	0,1	19702	26319
4	6	10	0,1	17885	33785
5	10	10	0,1	16613	Infinite
6	15	10	0,1	17452	Infinite
7	21	10	0,1	17451	Infinite
8	28	10	0,1	16934	Infinite
9	36	10	0,1	14145	Infinite
10	45	10	0,1	16150	27797
11	55	10	0,1	17023	28600
12	66	10	0,1	20542	Infinite
3	3	20	0,1	20542	Infinite
4	6	20	0,1	17393	Infinite
5	10	20	0,1	15878	Infinite
6	15	20	0,1	15578	Infinite
7	21	20	0,1	16092	Infinite
8	28	20	0,1	18771	Infinite
9	36	20	0,1	19060	Infinite
10	45	20	0,1	15545	Infinite
11	55	20	0,1	14103	Infinite
12	66	20	0,1	16935	Infinite
3	3	30	0,1	19702	26319
4	6	30	0,1	20351	Infinite
5	10	30	0,1	15708	Infinite
6	15	30	0,1	14373	Infinite
7	21	30	0,1	14047	Infinite
8	28	30	0,1	17326	Infinite
9	36	30	0,1	21239	Infinite
10	45	30	0,1	14885	Infinite
11	55	30	0,1	12134	Infinite
12	66	30	0,1	23005	Infinite
3	3	40	0,1	19702	26319
4	6	40	0,1	Infinite	Infinite

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
5	10	40	0,1	15152	Infinite
6	15	40	0,1	13570	Infinite
7	21	40	0,1	16481	Infinite
8	28	40	0,1	21194	Infinite
9	36	40	0,1	20445	Infinite
10	45	40	0,1	15575	Infinite
11	55	40	0,1	11719	Infinite
12	66	40	0,1	14912	Infinite
3	3	50	0,1	19702	26319
4	6	50	0,1	Infinite	Infinite
5	10	50	0,1	14976	Infinite
6	15	50	0,1	13623	Infinite
7	21	50	0,1	20450	Infinite
8	28	50	0,1	41472	Infinite
9	36	50	0,1	22808	Infinite
10	45	50	0,1	14444	Infinite
11	55	50	0,1	11352	Infinite
12	66	50	0,1	15328	Infinite
3	3	10	0,2	19702	26319
4	6	10	0,2	17885	33785
5	10	10	0,2	18616	Infinite
6	15	10	0,2	20702	25029
7	21	10	0,2	17719	96716
8	28	10	0,2	17414	25734
9	36	10	0,2	15931	Infinite
10	45	10	0,2	15214	Infinite
11	55	10	0,2	17468	29235
12	66	10	0,2	201542	Infinite
3	3	20	0,2	19702	26319
4	6	20	0,2	17393	Infinite
5	10	20	0,2	18189	Infinite
6	15	20	0,2	14861	21721
7	21	20	0,2	17954	Infinite
8	28	20	0,2	19141	Infinite

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
9	36	20	0,2	14677	Infinite
10	45	20	0,2	29229	Infinite
11	55	20	0,2	15258	Infinite
12	66	20	0,2	16395	Infinite
3	3	30	0,2	19702	26319
4	6	30	0,2	20335	Infinite
5	10	30	0,2	15062	Infinite
6	15	30	0,2	13360	Infinite
7	21	30	0,2	20234	Infinite
8	28	30	0,2	22945	Infinite
9	36	30	0,2	13436	Infinite
10	45	30	0,2	21704	Infinite
11	55	30	0,2	13867	Infinite
12	66	30	0,2	22648	Infinite
3	3	40	0,2	19702	26319
4	6	40	0,2	Infinite	Infinite
5	10	40	0,2	14066	Infinite
6	15	40	0,2	11488	Infinite
7	21	40	0,2	16923	Infinite
8	28	40	0,2	18860	Infinite
9	36	40	0,2	14887	Infinite
10	45	40	0,2	17744	Infinite
11	55	40	0,2	13214	Infinite
12	66	40	0,2	15323	Infinite
3	3	50	0,2	19702	26319
4	6	50	0,2	Infinite	Infinite
5	10	50	0,2	11212	Infinite
6	15	50	0,2	11158	Infinite
7	21	50	0,2	16745	Infinite
8	28	50	0,2	24779	Infinite
9	36	50	0,2	20794	Infinite
10	45	50	0,2	15370	Infinite
11	55	50	0,2	12969	Infinite
12	66	50	0,2	19838	Infinite

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
3	3	10	0,3	Infinite	Infinite
4	6	10	0,3	17885	33785
5	10	10	0,3	16636	29640
6	15	10	0,3	17452	Infinite
7	21	10	0,3	16262	Infinite
8	28	10	0,3	16012	Infinite
9	36	10	0,3	23136	Infinite
10	45	10	0,3	38055	30413
11	55	10	0,3	15954	Infinite
12	66	10	0,3	20542	Infinite
3	3	20	0,3	Infinite	Infinite
4	6	20	0,3	17393	Infinite
5	10	20	0,3	16496	Infinite
6	15	20	0,3	12930	Infinite
7	21	20	0,3	15097	Infinite
8	28	20	0,3	23533	Infinite
9	36	20	0,3	23943	Infinite
10	45	20	0,3	20421	Infinite
11	55	20	0,3	15823	Infinite
12	66	20	0,3	16935	Infinite
3	3	30	0,3	Infinite	Infinite
4	6	30	0,3	20335	Infinite
5	10	30	0,3	15275	Infinite
6	15	30	0,3	13041	Infinite
7	21	30	0,3	16331	Infinite
8	28	30	0,3	14889	Infinite
9	36	30	0,3	20342	Infinite
10	45	30	0,3	23741	Infinite
11	55	30	0,3	15798	Infinite
12	66	30	0,3	226448	Infinite
3	3	40	0,3	Infinite	Infinite
4	6	40	0,3	Infinite	Infinite
5	10	40	0,3	14222	Infinite
6	15	40	0,3	13224	Infinite



Lag Input	Maksimum Neuron	Maksimum Lapisan	$\alpha$	RMSE (pelatihan)	RMSE (pengujian)
7	21	40	0,3	27473	Infinite
8	28	40	0,3	16423	Infinite
9	36	40	0,3	19474	Infinite
10	45	40	0,3	20995	Infinite
11	55	40	0,3	14535	Infinite
12	66	40	0,3	15323	Infinite
3	3	50	0,3	Infinite	Infinite
4	6	50	0,3	Infinite	Infinite
5	10	50	0,3	13838	Infinite
6	15	50	0,3	14008	Infinite
7	21	50	0,3	22809	Infinite
8	28	50	0,3	15357	Infinite
9	36	50	0,3	20845	Infinite
10	45	50	0,3	21307	Infinite
11	55	50	0,3	15178	Infinite
12	66	50	0,3	19838	Infinite
3	3	10	0,4	19702	26319
4	6	10	0,4	19112	27760
5	10	10	0,4	19666	Infinite
6	15	10	0,4	17452	Infinite
7	21	10	0,4	16392	31854
8	28	10	0,4	13747	Infinite
9	36	10	0,4	16571	Infinite
10	45	10	0,4	22129	Infinite
11	55	10	0,4	20317	32062
12	66	10	0,4	15807	29850
3	3	20	0,4	19702	26319
4	6	20	0,4	19112	27760
5	10	20	0,4	15783	Infinite
6	15	20	0,4	12930	Infinite
7	21	20	0,4	14656	Infinite
8	28	20	0,4	13391	Infinite
9	36	20	0,4	18100	Infinite
10	45	20	0,4	22881	Infinite

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
11	55	20	0,4	16857	Infinite
12	66	20	0,4	35276	Infinite
3	3	30	0,4	19702	26319
4	6	30	0,4	19112	27760
5	10	30	0,4	12574	Infinite
6	15	30	0,4	13041	Infinite
7	21	30	0,4	20341	Infinite
8	28	30	0,4	14963	Infinite
9	36	30	0,4	26144	Infinite
10	45	30	0,4	14336	Infinite
11	55	30	0,4	14260	Infinite
12	66	30	0,4	29007	Infinite
3	3	40	0,4	19702	26319
4	6	40	0,4	19112	27760
5	10	40	0,4	12125	Infinite
6	15	40	0,4	13224	Infinite
7	21	40	0,4	20903	Infinite
8	28	40	0,4	18155	Infinite
9	36	40	0,4	20704	Infinite
10	45	40	0,4	13732	Infinite
11	55	40	0,4	14927	Infinite
12	66	40	0,4	15186	Infinite
3	3	50	0,4	19702	26319
4	6	50	0,4	19112	27760
5	10	50	0,4	12742	Infinite
6	15	50	0,4	14008	Infinite
7	21	50	0,4	18871	Infinite
8	28	50	0,4	16193	Infinite
9	36	50	0,4	33203	Infinite
10	45	50	0,4	13184	Infinite
11	55	50	0,4	14286	Infinite
12	66	50	0,4	13958	Infinite
3	3	10	0,5	19702	26319
4	6	10	0,5	18875	28021

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
5	10	10	0,5	17987	27873
6	15	10	0,5	15407	Infinite
7	21	10	0,5	39465	1,53E+93
8	28	10	0,5	15550	39465
9	36	10	0,5	17197	Infinite
10	45	10	0,5	17473	26611
11	55	10	0,5	16748	Infinite
12	66	10	0,5	18106	32343
3	3	20	0,5	19702	26319
4	6	20	0,5	18875	28021
5	10	20	0,5	17987	27873
6	15	20	0,5	13664	Infinite
7	21	20	0,5	14712	Infinite
8	28	20	0,5	14172	Infinite
9	36	20	0,5	14304	Infinite
10	45	20	0,5	15499	Infinite
11	55	20	0,5	15561	Infinite
12	66	20	0,5	20969	Infinite
3	3	30	0,5	19702	26319
4	6	30	0,5	18875	28021
5	10	30	0,5	17987	27873
6	15	30	0,5	13529	Infinite
7	21	30	0,5	19060	Infinite
8	28	30	0,5	13457	Infinite
9	36	30	0,5	13709	Infinite
10	45	30	0,5	14914	Infinite
11	55	30	0,5	15935	Infinite
12	66	30	0,5	13037	Infinite
3	3	40	0,5	19702	26319
4	6	40	0,5	18875	28021
5	10	40	0,5	17987	27873
6	15	40	0,5	11185	Infinite
7	21	40	0,5	14894	Infinite
8	28	40	0,5	14050	Infinite

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
9	36	40	0,5	13423	Infinite
10	45	40	0,5	15415	Infinite
11	55	40	0,5	14298	Infinite
12	66	40	0,5	12138	Infinite
3	3	50	0,5	19702	26319
4	6	50	0,5	18875	28021
5	10	50	0,5	17987	27873
6	15	50	0,5	11009	Infinite
7	21	50	0,5	18091	Infinite
8	28	50	0,5	16215	Infinite
9	36	50	0,5	19214	Infinite
10	45	50	0,5	14891	Infinite
11	55	50	0,5	14064	Infinite
12	66	50	0,5	12135	Infinite

**Tabel H 3 Uji Coba Terhadap Penambahan Selection Pressure  
0,6-1**

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
3	3	10	0,6	19702	26319
4	6	10	0,6	18875	28021
5	10	10	0,6	18377	27920
6	15	10	0,6	18129	24526
7	21	10	0,6	15426	Infinite
8	28	10	0,6	15551	27969
9	36	10	0,6	14173	Infinite
10	45	10	0,6	16627	26456
11	55	10	0,6	16501	30775
12	66	10	0,6	24692	31307
3	3	20	0,6	19702	26319
4	6	20	0,6	18875	28021
5	10	20	0,6	18377	27920
6	15	20	0,6	18129	24526
7	21	20	0,6	20823	Infinite
8	28	20	0,6	12599	Infinite
9	36	20	0,6	16522	Infinite
10	45	20	0,6	14851	Infinite
11	55	20	0,6	18207	Infinite
12	66	20	0,6	17815	Infinite
3	3	30	0,6	19702	26319
4	6	30	0,6	18875	28021
5	10	30	0,6	18377	27920
6	15	30	0,6	18129	24526
7	21	30	0,6	14629	Infinite
8	28	30	0,6	20167	Infinite
9	36	30	0,6	23018	Infinite
10	45	30	0,6	14469	Infinite
11	55	30	0,6	19840	Infinite
12	66	30	0,6	14001	Infinite
3	3	40	0,6	19702	26319

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
4	6	40	0,6	18875	28021
5	10	40	0,6	18377	27920
6	15	40	0,6	18129	24526
7	21	40	0,6	13217	Infinite
8	28	40	0,6	12371	Infinite
9	36	40	0,6	20985	Infinite
10	45	40	0,6	16470	Infinite
11	55	40	0,6	17904	Infinite
12	66	40	0,6	11994	Infinite
3	3	50	0,6	19702	26319
4	6	50	0,6	18875	28021
5	10	50	0,6	18377	27920
6	15	50	0,6	18129	24526
7	21	50	0,6	12824	Infinite
8	28	50	0,6	15614	Infinite
9	36	50	0,6	24876	Infinite
10	45	50	0,6	15969	Infinite
11	55	50	0,6	18935	Infinite
12	66	50	0,6	11867	Infinite
3	3	10	0,7	19702	26319
4	6	10	0,7	18875	28021
5	10	10	0,7	18377	27920
6	15	10	0,7	17744	23473
7	21	10	0,7	17761	23431
8	28	10	0,7	17696	27754
9	36	10	0,7	30930	Infinite
10	45	10	0,7	15035	29553
11	55	10	0,7	19597	33298
12	66	10	0,7	19597	33298
3	3	20	0,7	19702	26319
4	6	20	0,7	18875	28021
5	10	20	0,7	18377	27920
6	15	20	0,7	17744	23473
7	21	20	0,7	17761	23431

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
8	28	20	0,7	17696	27754
9	36	20	0,7	24819	Infinite
10	45	20	0,7	23966	Infinite
11	55	20	0,7	17255	Infinite
12	66	20	0,7	18658	Infinite
3	3	30	0,7	19702	26319
4	6	30	0,7	18875	28021
5	10	30	0,7	18377	27920
6	15	30	0,7	17744	23473
7	21	30	0,7	17761	23431
8	28	30	0,7	17696	27754
9	36	30	0,7	19470	Infinite
10	45	30	0,7	17200	Infinite
11	55	30	0,7	18426	Infinite
12	66	30	0,7	18482	Infinite
3	3	40	0,7	19702	26319
4	6	40	0,7	18875	28021
5	10	40	0,7	18377	27920
6	15	40	0,7	17744	23473
7	21	40	0,7	17761	23431
8	28	40	0,7	17696	27754
9	36	40	0,7	18495	Infinite
10	45	40	0,7	16207	Infinite
11	55	40	0,7	17534	Infinite
12	66	40	0,7	19327	Infinite
3	3	50	0,7	19702	26319
4	6	50	0,7	18875	28021
5	10	50	0,7	18377	27920
6	15	50	0,7	17744	23473
7	21	50	0,7	17761	23431
8	28	50	0,7	17696	27754
9	36	50	0,7	15858	Infinite
10	45	50	0,7	15003	Infinite
11	55	50	0,7	16836	Infinite

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
12	66	50	0,7	16890	Infinite
3	3	10	0,8	19843	25789
4	6	10	0,8	19279	27694
5	10	10	0,8	18670	28071
6	15	10	0,8	17744	23473
7	21	10	0,8	17761	23431
8	28	10	0,8	18902	25020
9	36	10	0,8	15967	Infinite
10	45	10	0,8	18749	30183
11	55	10	0,8	17351	31070
12	66	10	0,8	16253	Infinite
3	3	20	0,8	19843	25789
4	6	20	0,8	19279	27694
5	10	20	0,8	18670	28071
6	15	20	0,8	17744	23473
7	21	20	0,8	17761	23431
8	28	20	0,8	18902	25020
9	36	20	0,8	13963	Infinite
10	45	20	0,8	18414	Infinite
11	55	20	0,8	15041	Infinite
12	66	20	0,8	14292	Infinite
3	3	30	0,8	19843	25789
4	6	30	0,8	19279	27694
5	10	30	0,8	18670	28071
6	15	30	0,8	17744	23473
7	21	30	0,8	17761	23431
8	28	30	0,8	18902	25020
9	36	30	0,8	13275	Infinite
10	45	30	0,8	15106	Infinite
11	55	30	0,8	19145	Infinite
12	66	30	0,8	15294	Infinite
3	3	40	0,8	19843	25789
4	6	40	0,8	19279	27694
5	10	40	0,8	18670	28071



<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
6	15	40	0,8	17744	23473
7	21	40	0,8	17761	23431
8	28	40	0,8	18902	25020
9	36	40	0,8	13852	Infinite
10	45	40	0,8	14977	Infinite
11	55	40	0,8	12454	Infinite
12	66	40	0,8	29538	Infinite
3	3	50	0,8	19843	25789
4	6	50	0,8	19279	27694
5	10	50	0,8	18670	28071
6	15	50	0,8	17744	23473
7	21	50	0,8	17761	23431
8	28	50	0,8	18902	25020
9	36	50	0,8	17200	Infinite
10	45	50	0,8	16306	Infinite
11	55	50	0,8	11277	Infinite
12	66	50	0,8	18209	Infinite
3	3	10	0,9	19843	25789
4	6	10	0,9	19623	27222
5	10	10	0,9	19264	27204
6	15	10	0,9	19307	27154
7	21	10	0,9	18976	27642
8	28	10	0,9	Infinite	Infinite
9	36	10	0,9	18048	27552
10	45	10	0,9	19363	29099
11	55	10	0,9	18936	28257
12	66	10	0,9	19059	29231
3	3	20	0,9	19843	25789
4	6	20	0,9	19623	27222
5	10	20	0,9	19264	27204
6	15	20	0,9	19307	27154
7	21	20	0,9	18976	27642
8	28	20	0,9	Infinite	Infinite
9	36	20	0,9	18048	27552

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
10	45	20	0,9	19363	29099
11	55	20	0,9	18936	28257
12	66	20	0,9	19059	29231
3	3	30	0,9	19843	25789
4	6	30	0,9	19623	27222
5	10	30	0,9	19264	27204
6	15	30	0,9	19307	27154
7	21	30	0,9	18976	27642
8	28	30	0,9	Infinite	Infinite
9	36	30	0,9	18048	27552
10	45	30	0,9	19363	29099
11	55	30	0,9	18936	28257
12	66	30	0,9	19059	29231
3	3	40	0,9	19843	25789
4	6	40	0,9	19623	27222
5	10	40	0,9	19264	27204
6	15	40	0,9	19307	27154
7	21	40	0,9	18976	27642
8	28	40	0,9	Infinite	Infinite
9	36	40	0,9	18048	27552
10	45	40	0,9	19363	29099
11	55	40	0,9	18936	28257
12	66	40	0,9	19059	29231
3	3	50	0,9	19843	25789
4	6	50	0,9	19623	27222
5	10	50	0,9	19264	27204
6	15	50	0,9	19307	27154
7	21	50	0,9	18976	27642
8	28	50	0,9	Infinite	Infinite
9	36	50	0,9	18048	27552
10	45	50	0,9	19363	29099
11	55	50	0,9	18936	28257
12	66	50	0,9	19059	29231
3	3	10	1	19843	25789

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
4	6	10	1	19623	27222
5	10	10	1	19264	27204
6	15	10	1	19307	27154
7	21	10	1	19368	27157
8	28	10	1	19323	27392
9	36	10	1	19356	27321
10	45	10	1	19527	26539
11	55	10	1	19562	26510
12	66	10	1	19578	26619
3	3	20	1	19843	25789
4	6	20	1	19623	27222
5	10	20	1	19264	27204
6	15	20	1	19307	27154
7	21	20	1	19368	27157
8	28	20	1	19323	27392
9	36	20	1	19356	27321
10	45	20	1	19527	26539
11	55	20	1	19562	26510
12	66	20	1	19578	26619
3	3	30	1	19843	25789
4	6	30	1	19623	27222
5	10	30	1	19264	27204
6	15	30	1	19307	27154
7	21	30	1	19368	27157
8	28	30	1	19323	27392
9	36	30	1	19356	27321
10	45	30	1	19527	26539
11	55	30	1	19562	26510
12	66	30	1	19578	26619
3	3	40	1	19843	25789
4	6	40	1	19623	27222
5	10	40	1	19264	27204
6	15	40	1	19307	27154
7	21	40	1	19368	27157

<b>Lag Input</b>	<b>Maksimum Neuron</b>	<b>Maksimum Lapisan</b>	<b><math>\alpha</math></b>	<b>RMSE (pelatihan)</b>	<b>RMSE (pengujian)</b>
8	28	40	1	19323	27392
9	36	40	1	19356	27321
10	45	40	1	19527	26539
11	55	40	1	19562	26510
12	66	40	1	19578	26619
3	3	50	1	19843	25789
4	6	50	1	19623	27222
5	10	50	1	19264	27204
6	15	50	1	19307	27154
7	21	50	1	19368	27157
8	28	50	1	19323	27392
9	36	50	1	19356	27321
10	45	50	1	19527	26539
11	55	50	1	19562	26510
12	66	50	1	19578	26619